

openCARP CDE - Konzept und Implementation einer
kollaborativen Softwareentwicklungsumgebung

Jochen Klar

Workshop: Forschungssoftware managen, 16.9.2022

openCARP

- openCARP ist ein offener kardialer Elektrophysiologie-Simulator für In-silico-Experimente.
- Der Quellcode ist öffentlich und die Software ist für nicht-kommerzielle Zwecke frei nutzbar.
- openCARP wird in enger Zusammenarbeit mit der Community entwickelt und weltweit genutzt.
- Im DFG Projekt **Sustainable Lifecycle Management for Scientific Software (SuLMaSS)**, 2018-2022, wurde eine nachhaltige Softwareentwicklungsumgebung für openCARP aufgebaut.



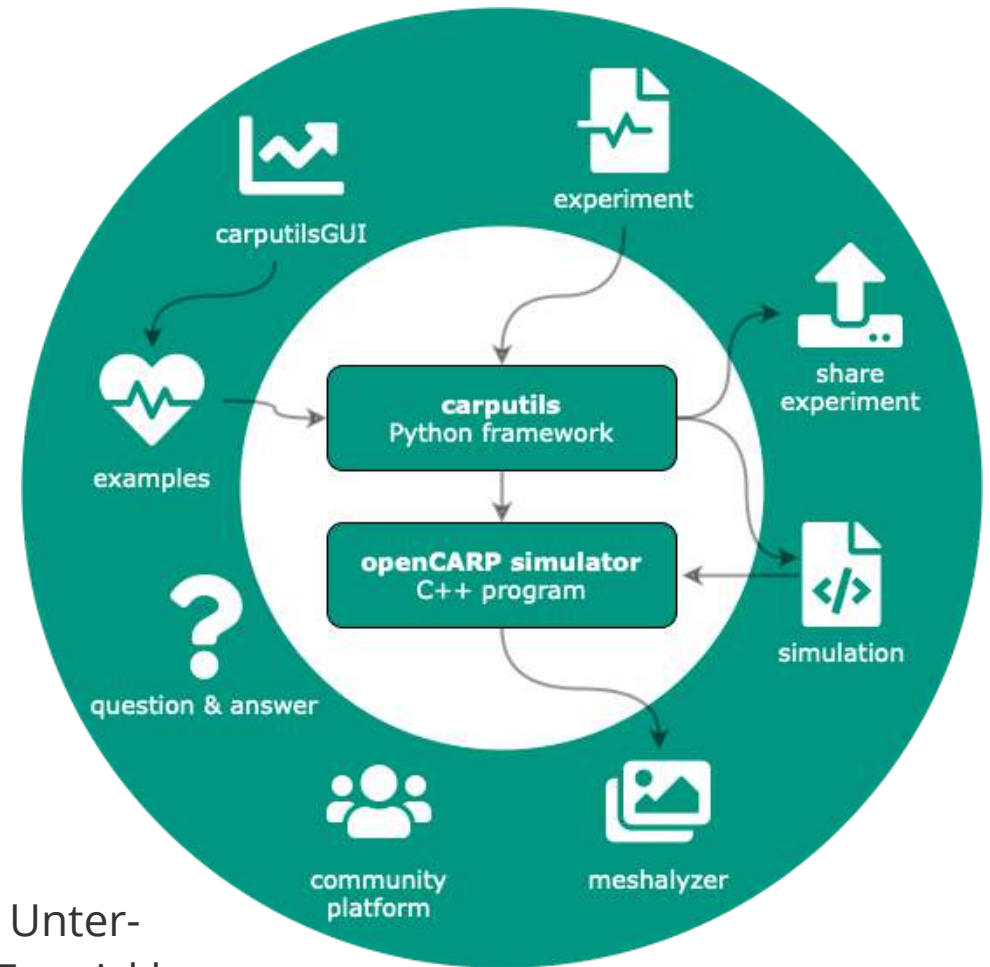
Gernot Plank, Axel Loewe, Aurel Neic, Christoph Augustin, Yung-Lin Huang, Matthias A.F. Gsell, Elias Karabelas, Mark Nothstein, Anton J. Prassl, Jorge Sánchez, Gunnar Seemann, Edward J. Vigmond: **The openCARP simulation environment for cardiac electrophysiology**. Computer Methods and Programs in Biomedicine, Volume 208, 2021, 106223. <https://doi.org/10.1016/j.cmpb.2021.106223>

Felix Bach, Jochen Klar, Axel Loewe, Jorge Sánchez, Gunnar Seemann, Yung-Lin Huang, Robert Ulrich: **The OpenCARP CDE: Concept for and Implementation of a Sustainable Collaborative Development Environment for Research Software**. Bausteine Forschungsdatenmanagement, Nr. 1 (März). German:64-84. (2022). <https://doi.org/10.17192/bfdm.2022.1.8368>

The openCARP ecosystem

- Der **openCARP** Simulator ist in C++ geschrieben.
- Das **carputils** Framework ist in Python geschrieben und Ermöglicht die Erstellung von Experimenten einschließlich Pre- und Postprocessing.
- **carputilsGUI** ist ein Web-basiertes Interface zu carputils und openCARP.
- **meshalyzer** ist ein graphisches Programm um zeitabhängige Daten auf 3D Meshes darzustellen.
- **meshtool** ist Kommandozeilentool zur Bearbeitung von volumetrischen Meshes.
- Zahlreiche **Examples** helfen bei dem Einstieg in die Arbeit mit openCARP und carputils.

Die **openCARP CDE** dient als **Community-Platform** zur Unterstützung der Nutzenden und der Entwicklerinnen und Entwickler.

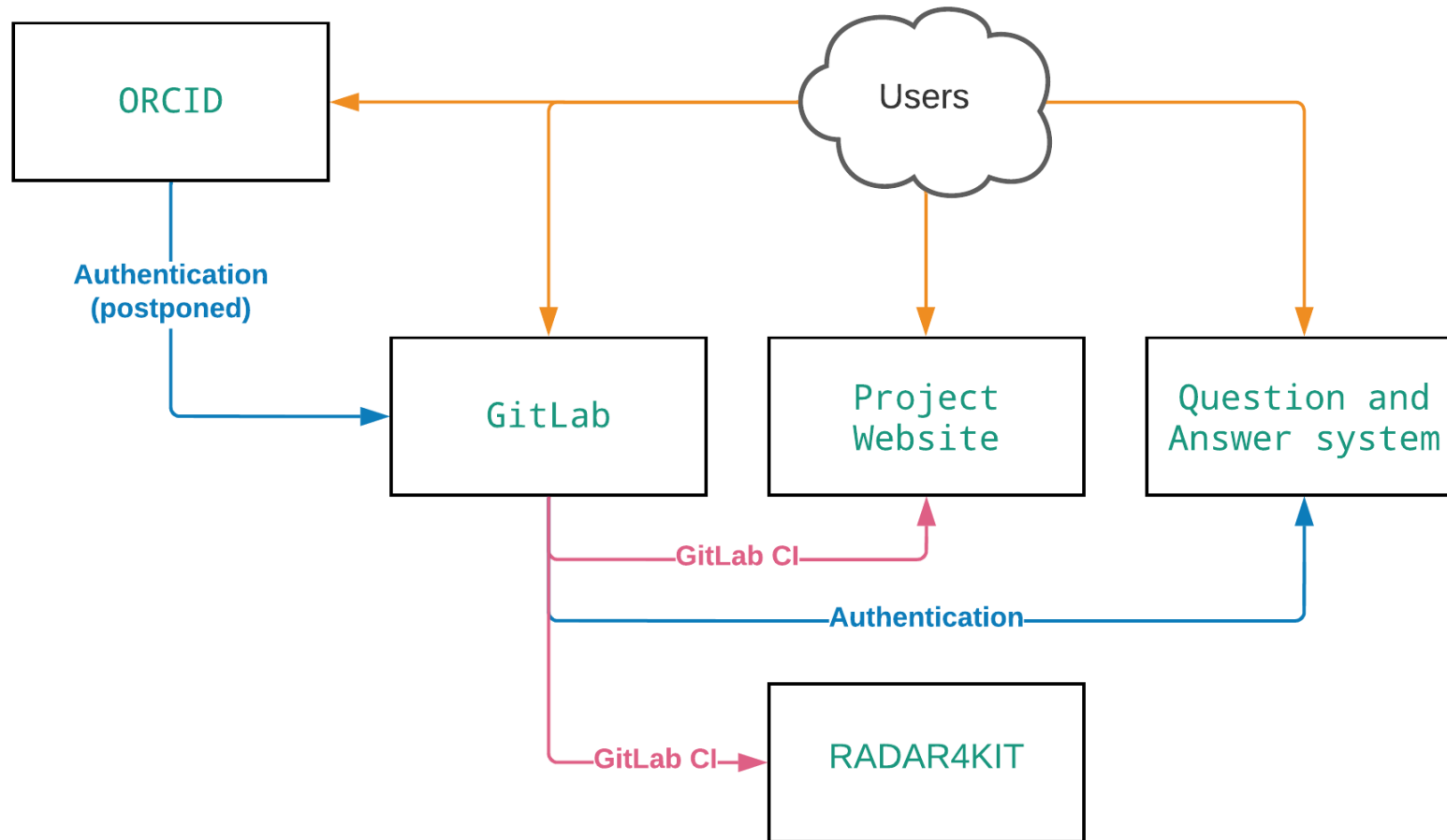


openCARP CDE

- openCARP nutzt eine eigene **GitLab** Instanz: git.opencarp.org
 - Üblicher GitHub/Lab Workflow: Issues, Branches, Pull/Merge Requests, Releases
 - Offene Registrierung, Login mit GitHub / GitLab.com Konten, ORCID Login geplant
 - GitLab CI/CD für Continuous Integration, Delivery, Deployment
 - Integrierte Docker-Registry: docker.opencarp.org
- Projektwebseite openCARP.org auf Basis von **GRAV**, einem Flat-File CMS
 - Synchronisation zum Gitlab über **Git-Sync Plugin**
- Frage-und-Antwort-Plattform **openCARP Q&A** auf Basis von **Question2Answer (Q2A)**
 - Authorisierung über GitLab via OAuth2
- **RADAR4KIT** für DOI, Publikation und Langzeitarchivierung
- Einfacher Webspaces für statische Assets

Infrastruktur: 2 virtuelle Maschinen, gehostet am **Steinbuch Centre for Computing (SCC)** des KIT

openCARP CRE



GitLab CI/CD Pipelines

- Erstellung von **Binärpaketen** (deb, rpm, pkg, ...)
- Erstellung eines **Docker-Images**
- Erstellen der **doxygen** Dokumentation als html
- Erstellen der **User-Dokumentation** aus LaTeX als pdf
- Erstellen der **Parameter-Dokumentation** aus XML in html
- Durchführung von **Regressionstests** und Erstellung eines Reports in html
- Erstellung eines **DataCite XML** nach Kernel 4.4
- **Synchronisation** von Inhalten des Git-Repo mit der Webpage
 - Im GRAV CMS wird auf bestimmten Seiten eine **pipeline** (z.B. **opencarp**) und eine **source** (z.B. **LICENSE.md**) definiert.
 - Die CI sucht die Seite für **pipeline** und überträgt die Informationen aus **source**.

Für **master/tags**: Upload der Artefakte auf den Webservice / die Docker-Registry.

GitLab CI/CD Pipelines (Releases)

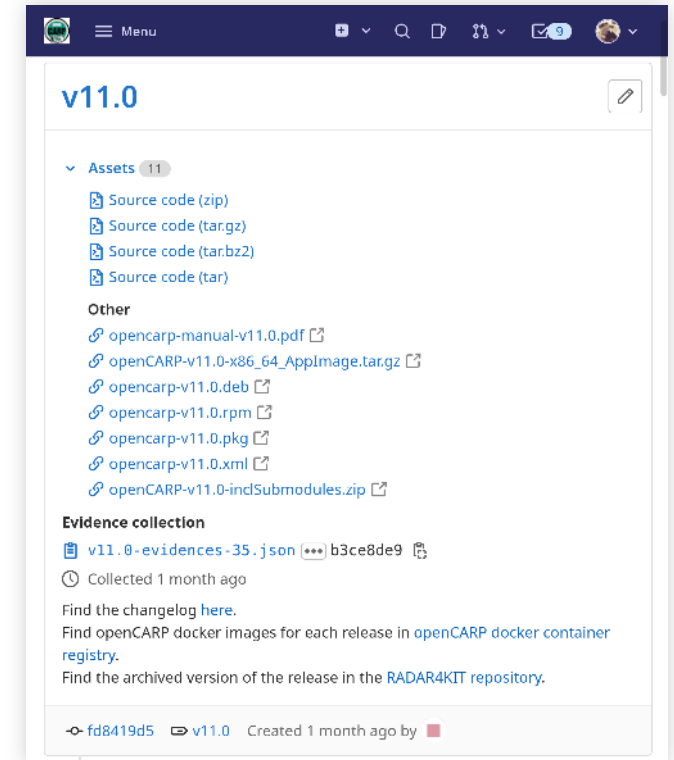
🚀 Releases werden erzeugt, indem `tags` in das Repo gepushed werden.

Zu den Releases gehören eine Reihe von **Assets**:

- **Source Code** (zip, tar.gz, tar.bz2, tar)
- **User-Dokumentation** (pdf)
- **Binärpakete** (appimage, deb, rpm, pkg)
- **DataCite XML**
- Source Code von **Submodulen**

In der CI werden dann zusätzliche Schritte durchgeführt:

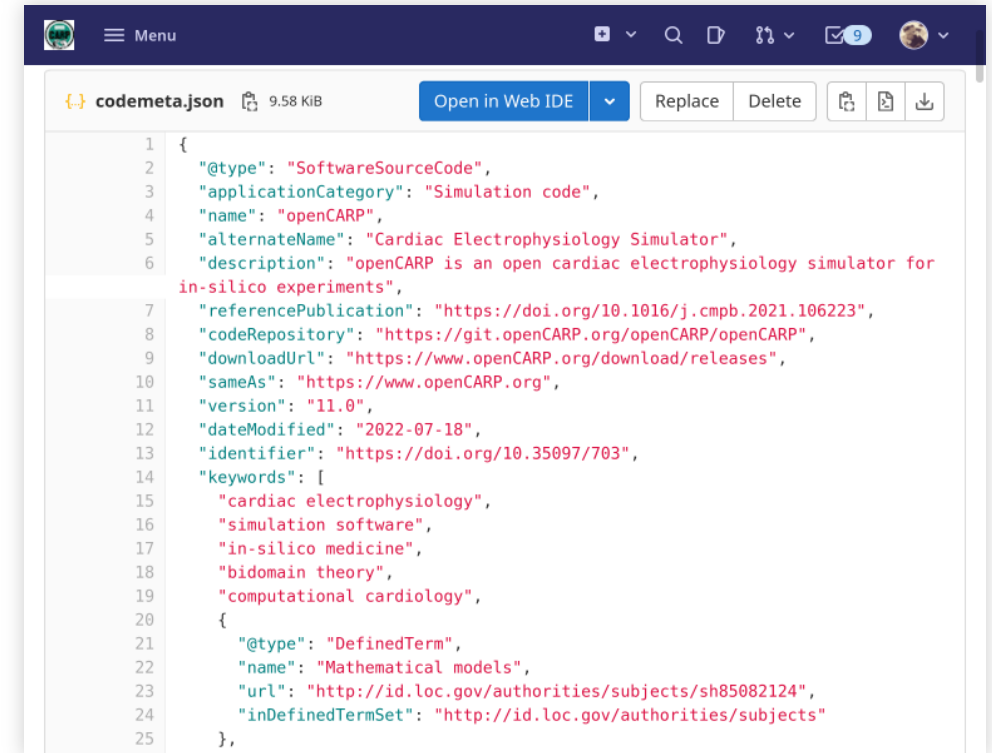
- Erstellung des **Releases** in GitLab per API mit Links zu den Assets.
- Erstellung eines Datensatzes in **RADAR4KIT** und Upload der Assets.
- Erstellung eines **bagpack** (`bagit` mit DataCite Metadaten) der extern abgespeichert wird.



Metadaten

Die benötigten Metadaten werden im Git-Repo gepflegt:

- Aktuelle Version:
 - `CONTRIBUTORS.yml` (für *authors*)
 - `METADATA.yml` (andere Metadaten)
- Zukünftige Version:
 - `codemeta.json` mit allen Metadaten
 - CodeMeta basiert auf schema.org
 - Mapping auf CFF, DataCite und RADAR
 - Zusätzliche `CITATION.cff` Datei
 - Synchronisation über `pre-commit` hooks und CI



```
1 {
2   "@type": "SoftwareSourceCode",
3   "applicationCategory": "Simulation code",
4   "name": "openCARP",
5   "alternateName": "Cardiac Electrophysiology Simulator",
6   "description": "openCARP is an open cardiac electrophysiology simulator for
7   in-silico experiments",
8   "referencePublication": "https://doi.org/10.1016/j.cmpb.2021.106223",
9   "codeRepository": "https://git.openCARP.org/openCARP/openCARP",
10  "downloadUrl": "https://www.openCARP.org/download/releases",
11  "sameAs": "https://www.openCARP.org",
12  "version": "11.0",
13  "dateModified": "2022-07-18",
14  "identifier": "https://doi.org/10.35097/703",
15  "keywords": [
16    "cardiac electrophysiology",
17    "simulation software",
18    "in-silico medicine",
19    "bidomain theory",
20    "computational cardiology",
21  ],
22  {
23    "@type": "DefinedTerm",
24    "name": "Mathematical models",
25    "url": "http://id.loc.gov/authorities/subjects/sh85082124",
26    "inDefinedTermSet": "http://id.loc.gov/authorities/subjects"
27  }
28 }
```

Es wird ein zusätzlicher Schritt in der CI nötig, der **Version**, **Veröffentlichungsdatum** und **DOI** vor dem Release in `codemeta.json` und `CITATION.cff` einfügt.

Archivierung und Publikation in RADAR4KIT



1. Ein Tag `pre-v12.0` wird in das Repository gepushed.
2. Die CI erzeugt über die API zunächst einen leeren Datensatz in RADAR4KIT. Über die API wird versucht den Datensatz zum Review frei zu geben. Da keine validen Metadaten vorliegen, schlägt dies aber fehl, eine DOI wird aber schon reserviert.
3. Mit der DOI, der RADAR-ID, Version und Datum wird `codemeta.json` aktualisiert.
4. Die Änderung wird in das Repo comitted und ein neuer Tag `v12.0` wird erzeugt.
5. Die Release-CI läuft wie beschrieben, der Datensatz wird aber nicht neu in RADAR erzeugt, sondern mit der ID aus dem `codemeta.json` wird der bestehende, leere Datensatz upgedated.
6. Die Assets werden in RADAR4KIT hoch geladen.
7. Abschließend wird der Datensatz nochmal manuell in RADAR4KIT überprüft und dann publiziert.

Bis auf den letzten Schritt erfolgt der Prozess vollautomatisch.

Archivierung und Publikation in RADAR₄KIT

DE EN LOGIN →

RADAR 4 KIT

You are here: [openCARP \(v11.0\)](#)

Dataset: openCARP (v11.0)

RADAR Metadata | Content | Statistics | Technical Metadata

Alternate identifier: (URL) <https://opencarp.org/download/releases>

Related identifier: (Is New Version Of) [10.35097/389](https://doi.org/10.35097/389) - DOI
(Is Documented By) [10.1016/j.cmpb.2021.106223](https://doi.org/10.1016/j.cmpb.2021.106223) - DOI

Creator/Author: openCARP consortium
Christoph Augustin [Medical University of Graz] <https://orcid.org/0000-0001-6341-4014>
Luca Azzolin [Karlsruhe Institute of Technology (KIT)] <https://orcid.org/0000-0002-2919-4617>
Jason Bayer [IHU Liryc] <https://orcid.org/0000-0003-0269-3499>
Martin Bishop
Patrick M. Boyle [University of Washington] <https://orcid.org/0000-0001-9048-1239>
Federica Caforio [Medical University of Graz] <https://orcid.org/0000-0002-2637-0195>
Fernando Campos
Caroline Mendonca Costa
Andrew Crozier
Thomas Fastl
Karl Gillette [Medical University of Graz] <https://orcid.org/0000-0002-0420-5375>
Matthias Gsell [Medical University of Graz] <https://orcid.org/0000-0001-7742-8193>
Marie Houillon [Karlsruhe Institute of Technology (KIT)] <https://orcid.org/0000-0002-5584-0233>
Yung-Lin (Cary) Huang [University of Freiburg] <https://orcid.org/0000-0003-4307-3907>
Kristian Gregorius Husted [Simula Research Laboratory] <https://orcid.org/0000-0001-6485-4206>
Elias Karabelas [University of Graz] <https://orcid.org/0000-0002-4496-1933>
Elias Karabelas [Medical University of Graz] <https://orcid.org/0000-0002-4496-1933>
Jonathan Krauß [Karlsruhe Institute of Technology (KIT)]
Avel Loewe [Karlsruhe Institute of Technology (KIT)] <https://orcid.org/0000-0002-2487-4744>
Laura Marx [Medical University of Graz] <https://orcid.org/0000-0002-6555-507X>
Aurel Neic [Medical University of Graz] <https://orcid.org/0000-0000-0000-0000>
Mark Nothstein [Karlsruhe Institute of Technology (KIT)]
Gernot Plank [Medical University of Graz] <https://orcid.org/0000-0002-7380-6908>
Anton Prasaal [Medical University of Graz] <https://orcid.org/0000-0002-1920-1377>
Jorge Sanchez [Karlsruhe Institute of Technology (KIT)] <https://orcid.org/0000-0002-0824-2691>
Gunnar Seemann [University of Freiburg] <https://orcid.org/0000-0001-7111-7992>
Joshua Steyer [Karlsruhe Institute of Technology (KIT)]
Jorge Sánchez [Karlsruhe Institute of Technology (KIT)] <https://orcid.org/0000-0002-0824-2691>
Nico Tippmann [Megware GmbH]
Ed Yigmond [University of Bordeaux] <https://orcid.org/0000-0003-1388-3589>

DOI: [10.35097/703](https://doi.org/10.35097/703)
Publication date: 2022-07-19

Download Dataset
DOWNLOAD (1.3 GB)

Download Metadata
RADAR | **DOWNLOAD**

Statistics
64 Views | 2 Downloads

Rights statement for the dataset
This work is licensed under ACADEMIC PUBLIC LICENSE (openCARP, v1.0)

Cite Dataset
openCARP consortium; Christoph Augustin; Luca Azzolin; et al. (2022): openCARP (v11.0). Karlsruhe Institute of Technology (KIT). DOI: [10.35097/703](https://doi.org/10.35097/703)

- RADAR4KIT ist ein disziplinübergreifendes Forschungsdatenrepositorium für die Archivierung und Publikation von Forschungsdaten aus wissenschaftlichen Projekten von Forschenden des KIT.
- Unter Forschungsdaten werden hierbei digitale Daten verstanden, die im Forschungsprozess entstehen, also auch Forschungssoftware.
- Datenpakete sind für mind. 25 Jahre gespeichert und öffentlich verfügbar, permanent und eindeutig identifizierbar und zitierbar
- RADAR4KIT basiert auf dem generischen Cloud-Dienst **RADAR** von **FIZ Karlsruhe**.

openCARP-CI

Die von der CI genutzten Python-Scripte können frei nachgenutzt werden:

<https://git.opencarp.org/openCARP/openCARP-CI>

```
# Installation
pip install git+https://git.opencarp.org/openCARP/openCARP-CI.git

# Enthaltene Scripte
create_bag           # erstellt das BagIt Paket
create_bagpack       # erstellt das Backpack Paket
create_cff           # erstellt CITATION.cff aus codemeta.json
create_datacite      # erstellt datacite.xml aus codemeta.json
create_radar         # erstellt/updated ein Datensatz in RADAD und uploaded Assets
create_release       # erstellt ein Release in GitLab (mit Assets)
prepare_radar        # erstellt ein leeres Dataset in RADAR um eine DOI zu erhalten
prepare_release      # updated codemeta.json mit "version" und "dateModified"
run_bibtex_pipeline  # updated eine GRAV Seite mit Informationen aus einem Bibtex file
run_docstring_pipeline # updated eine GRAV Seite mit einer Docstring Dokumentation
run_markdown_pipeline # updated eine GRAV Seite mit Informationen aus einem Markdown file
```

Steuerung erfolgt über Kommandozeilenargumente oder Umgebungsvariablen.



Vielen Dank für die Aufmerksamkeit!

mail@jochenklar.de / [@jochenklar](https://www.instagram.com/jochenklar) / jochenklar.de

info@opencarp.org / opencarp.org / git.opencarp.org