

HERMES Kickoff Workshop 2021-11-12

Code management in **openCARP** & publication on **RADAR4KIT**

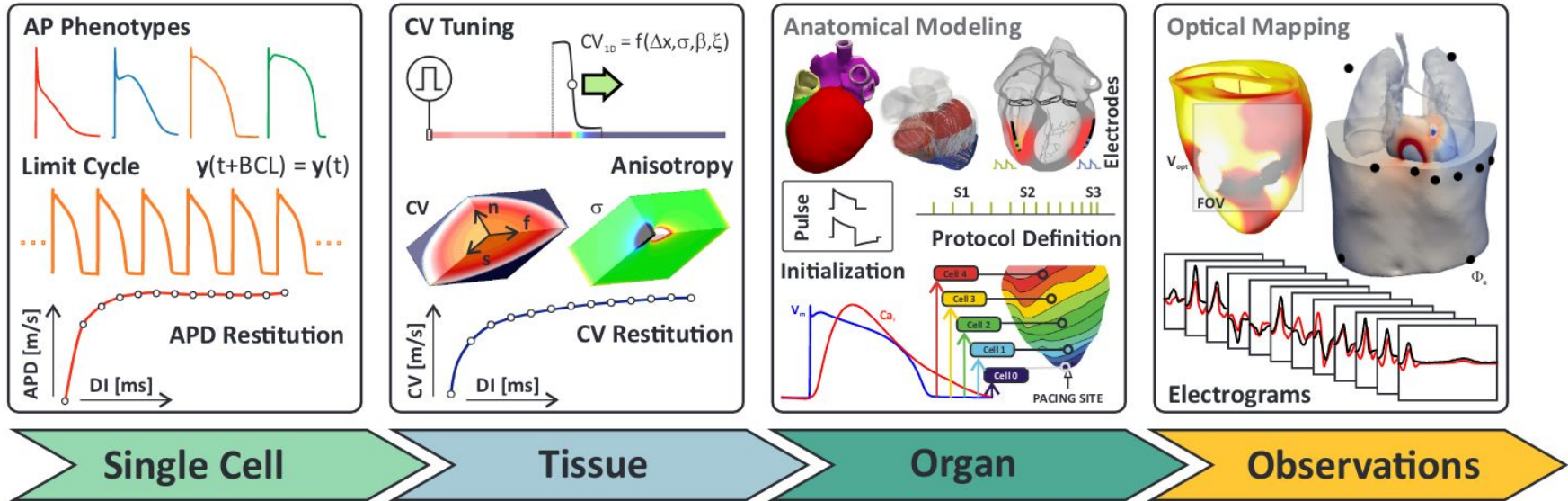


Overview

- openCARP software
 - What is it
 - CDE infrastructure
- RADAR4KIT repository
 - What is it?
 - RADAR4KIT in the RDM services landscape at KIT
 - Overview and functions
- Publication of openCARP software versions in RADAR4KIT and archiving in bwDataArchive

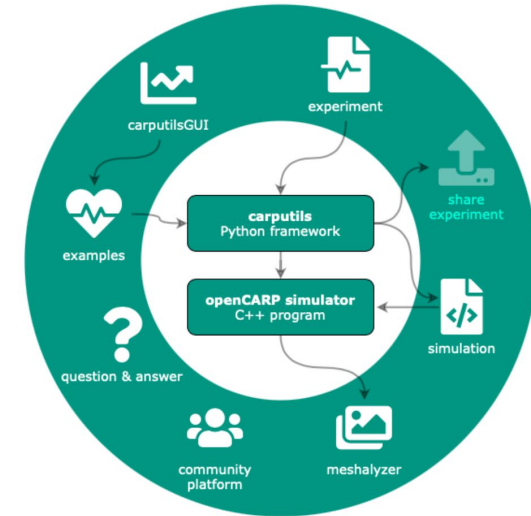
openCARP - What is it?

- **simulation** environment for the **computational cardiac electrophysiology** community



openCARP - What is it?

- **simulation** environment for the **computational cardiac electrophysiology** community
 - **Focus:** Usability and performance
 - extensive **training** material
 - includes Python-based **carputils** framework for **developing** and sharing **simulation pipelines** that are reproducible
 - community platform as a basis for a **vivid user community**
 - lively and heterogenous **developer community**



openCARP CDE framework

...consisting of

- a collaborative development environment (gitLab based)
- a DevOps platform
- an interactive project webpage
- tailored continuous integration pipelines
- automated version publishing

Purpose: Comprehensive yet **lightweight infrastructure for research software projects**. Could be a suitable solution for a wide range of research software projects to support sustainability

Paper on openCARP CDE developed in the project SuLMaSS to be published in *Bausteine FDM* soon

RADAR4KIT - what is that?

- Repository for RD
 - generic (not discipline specific) - “institutional”
 - “long tail of science”, discipline specific repos may have more impact!
 - Metadata in DataCite Scheme (Standard) and RADAR Scheme
 - discipline specific metadata is also possible (XML)
- Helps scientists to fulfill DFG-policies for good scientific practice (GSP)
 - Management of data and metadata
 - independent publication of RD - FAIR, with DOI
 - Long-term archiving

RDM-Services at KIT

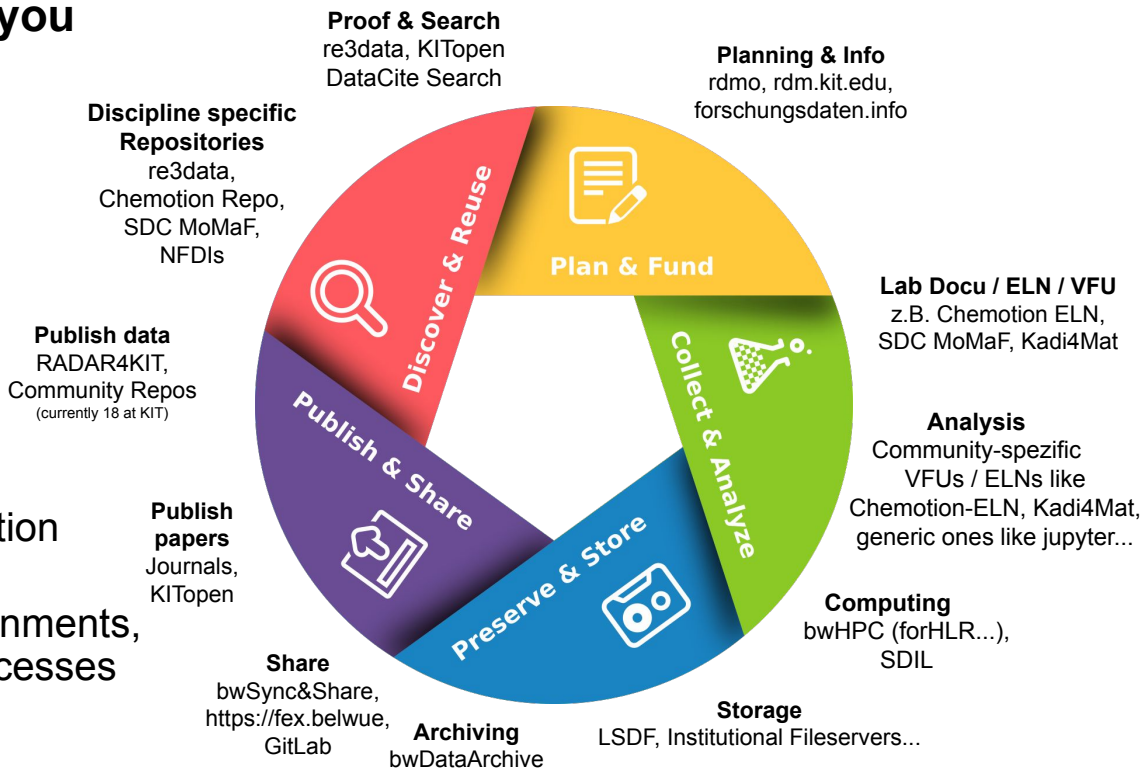
Serviceteam RDM@KIT supports you

Support: rdm.kit.edu,
contact@rdm.kit.edu

Required by scientists

- modern RD-Repository
- easy usability
- Roles and rights
- data sharing
- easy and independent data publication (FAIR with DOI and Metadata)
- Good integration in research environments, KIT-infrastructure und scientific processes

[more? contact@rdm.kit.edu](mailto:contact@rdm.kit.edu)



RADAR4KIT in the RD cycle

Functions

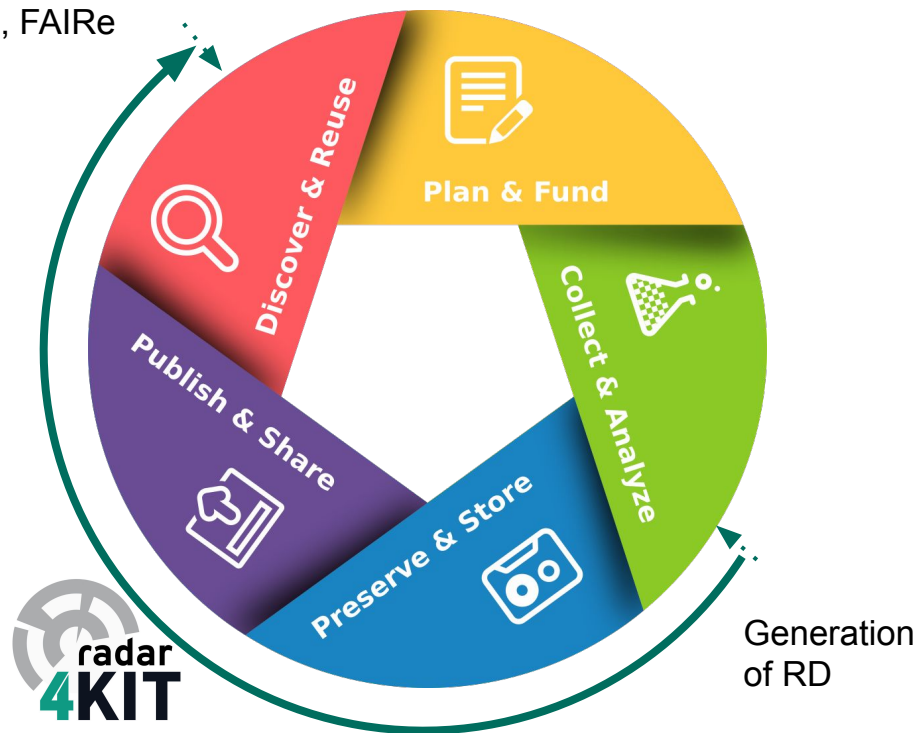
1) prepare Datasets:

- upload files
- describe data with metadata
- relate (e.g. by DOI) datasets and objects (e.g. publications)
- manage, collaborate, curate, share
 - e.g. in working group
 - for reviewers (temporarily)

2) final Datasets

- a) **publish** or
- b) **archive** (limited access)

Reuse, FAIR
RD



background: RADAR4KIT software

- **RADAR** (Repo-SW & service) developed in DFG-Project (2013-2016)
ReseArch DATA Repository



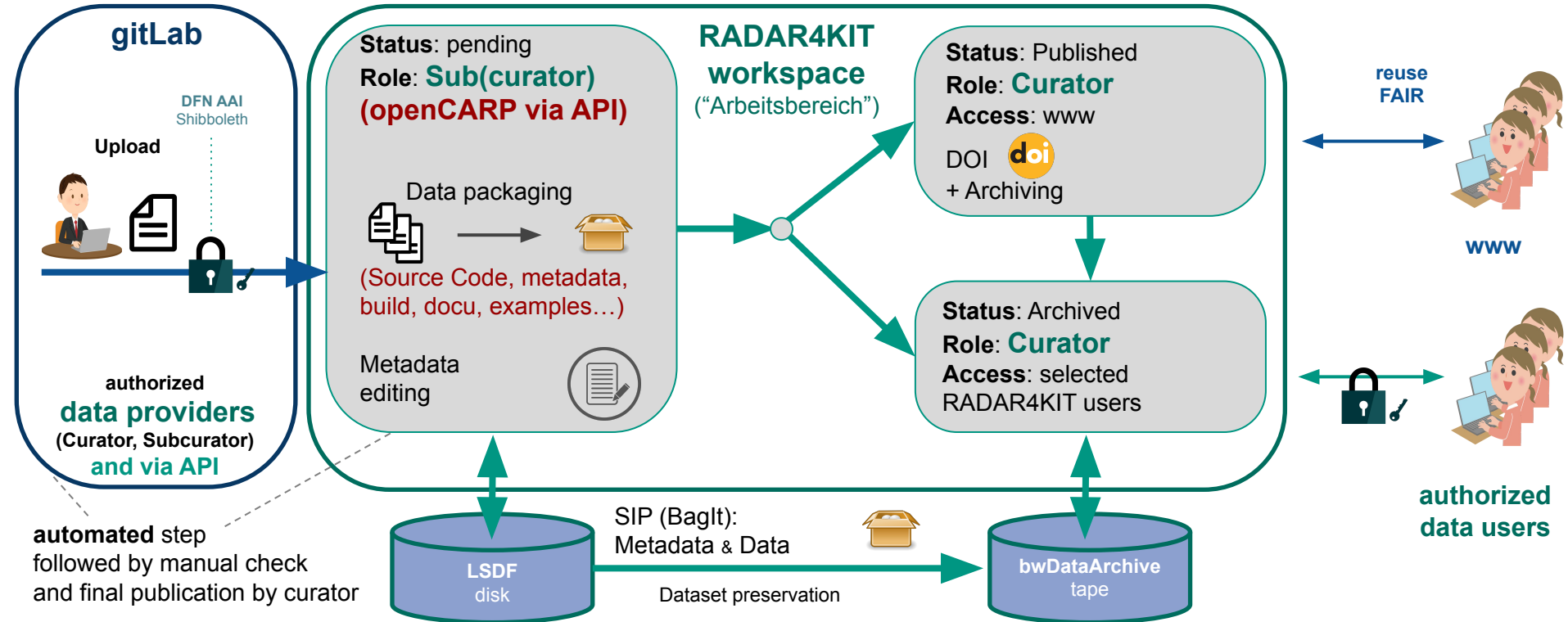
generic RD-Repo

Interdisciplinary project consortium
of infrastructure and community partners

- main infrastructure at KIT SCC
- since 2017 productive operation as commercial service by FIZ Karlsruhe
<https://radar-service.eu> -
- sustainable SW further development by RADAR consortium
- **RADAR4KIT**
 - own exclusive instance for KIT (all IT at SCC) - free for scientists
 - sustainable development and maintenance by FIZ (& close cooperation)
 - integration in KIT-systems possible using interfaces (API, OAI-PMH)



openCARP software publication in RADAR4KIT





BWData
ARCHIVE

rda.kit.edu

Architecture bwDataArchive



End users

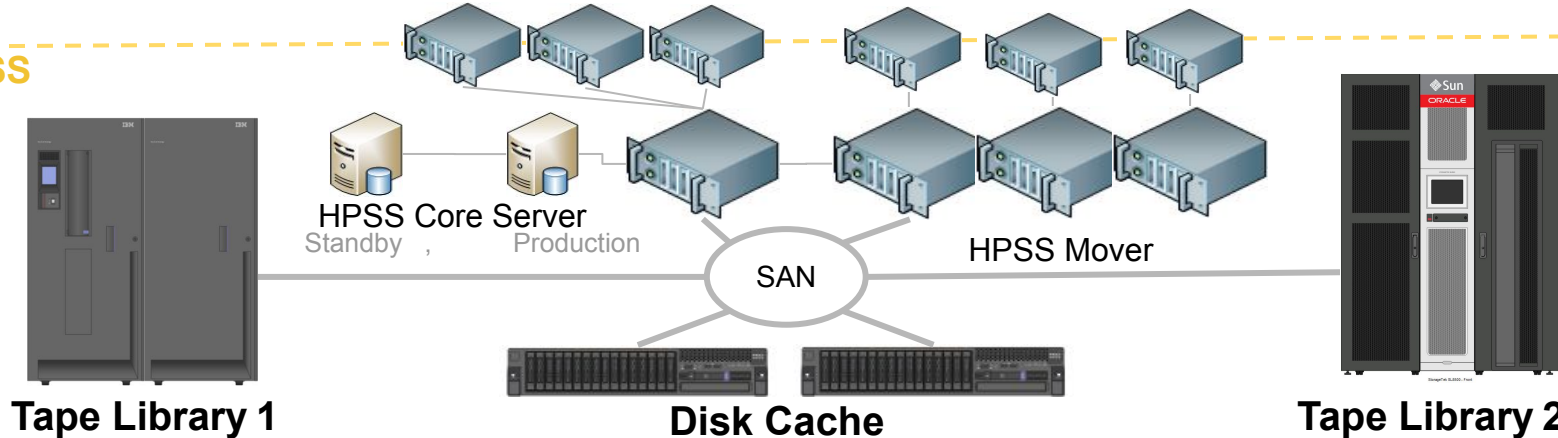


Services e.g. repositories



Access-Server (sftp) with HPSS mounted

IBM HPSS



Thank you for your attention



gitLab for openCARP

- GitLab allows creating an infrastructure to host and maintain the main openCARP git repository as well as a number of additional repositories for the different components of the openCARP environment. It offers a web interface supporting the main functionality of git (commits, branches, tags) as well as a web IDE and manages user accounts and access permissions to the repositories.
- Building upon these functions, it supports a state-of-the art software development workflow including merge requests, issue tracking, code review and release management. This includes user notifications and various tools for the evaluation and visualization of the development process.
- Integrated within GitLab is a powerful continuous integration (CI) framework, which allows performing automatic tests, generating documentation and other tasks, whenever a new commit is pushed to the repository. We heavily rely on this feature to connect and integrate the different components. The CI functionality is described in more detail in Section 2.2.
- GitLab ships with its own docker registry (in our case docker.opencarp.org). Docker⁴ uses container-based virtualization to encapsulate the whole system environment of an application (except the kernel of the operating system). Using Docker, we can create images for openCARP, which include all prerequisites and can therefore be installed with much less friction than would be the case for distribution of the source code alone.

host and maintain openCARP git repo,
webinterface with IDE, user mgmnt

workflow incl. merges, issues, review,
release with cool extras

CI integrated - auto-testing, docu-gen

Docker integration - make openCARP
installation easier

CI for openCARP (1/3)

- Build `deb` and `rpm` packages for Debian resp. CentOS based Linux distributions, AppImages for Linux-based systems, as well as `pkg` packages for macOS. These binary packages allow for an automatic installation of openCARP and all its system prerequisites.
- Build a Docker image from the openCARP source and upload it to the openCARP docker registry. Docker allows for a frictionless deployment of openCARP on any System which can run Docker (Linux, macOS, Windows).
- Build the Doxygen⁵ documentation of openCARP and copy it to a public directory on opencarp.org⁶. Doxygen automatically creates a browsable documentation from comments (so-called docstrings) in the source code.
- Build the user manual as pdf and copy it to a public directory on opencarp.org.

Package builds for different OS

Docker images builds

Docu creation

User manual creation

CI for openCARP (2/3)

- Build a browsable interface for the different parameters of openCARP and copy it to a public directory on opencarp.org⁷. This and the artifacts mentioned in the two points above are available per git branch and tags and thus allow users to go back to the documentation of any specific version.
- Build a browseable report interface for the available regression tests and copy the report to the public directory on opencarp.org⁸.
- Generate a DataCite record from structured metadata files in the code repository (see Section 2.6) and copy it to a public directory on opencarp.org.
- Create releases for certain versions of the code in GitLab. On a technical level, releases are tags in git, which have certain assets assigned to them. In our case, these are the different binary distributions of openCARP. GitLab creates a release page where those assets are accessible, as well as a tarball of the repository (at the release commit).
- Create a dataset in the long-term archive RADAR4KIT⁹ and upload the release assets (see Section 2.6).
- Create a backpack archive with the release assets and the DataCite metadata file (see Section 2.6).

create browsable interface for
openCARP params

build reports

build datacite record

create releases with
binary (tar)

create dataset in RADAR4KIT
(via API)

create backpack (for optional
archiving)

CI for openCARP (3/3)

- Synchronize the content of the opencarp.org GRAV content management system with structured metadata files from the code repository. Specific pages in the CMS are tagged with `pipeline` and `source` metadata entries. For openCARP, if a page has the pipeline `opencarp`, the CI inserts the content of the file given as source into the metadata of the page. Its content can then be used as structured data by the CMS, e.g. to use the contributors list from the repository to display on the public web page¹⁰ (see Section 2.3 for more information on the GRAV CMS).

sync GRAV CMS with
pipelines, source and metadata

Thank you!

Where to learn more about project HERMES?



Stephan Druskat, DLR, PI, [@stdruskat](#)



Oliver Bertuch, FZJ, PI, [@poi_ki_lo_therm](#)



Guido Juckeland, HZDR, PI, [@GuidoJuckeland](#)



Oliver Knodel, HZDR, [@olikno1](#)



Tobias Schlauch, DLR, [@TobiasSchlauch](#)

- Find us on [Twitter](#)
- Write an email to team@software-metadata.pub
- Go to software-metadata.pub