



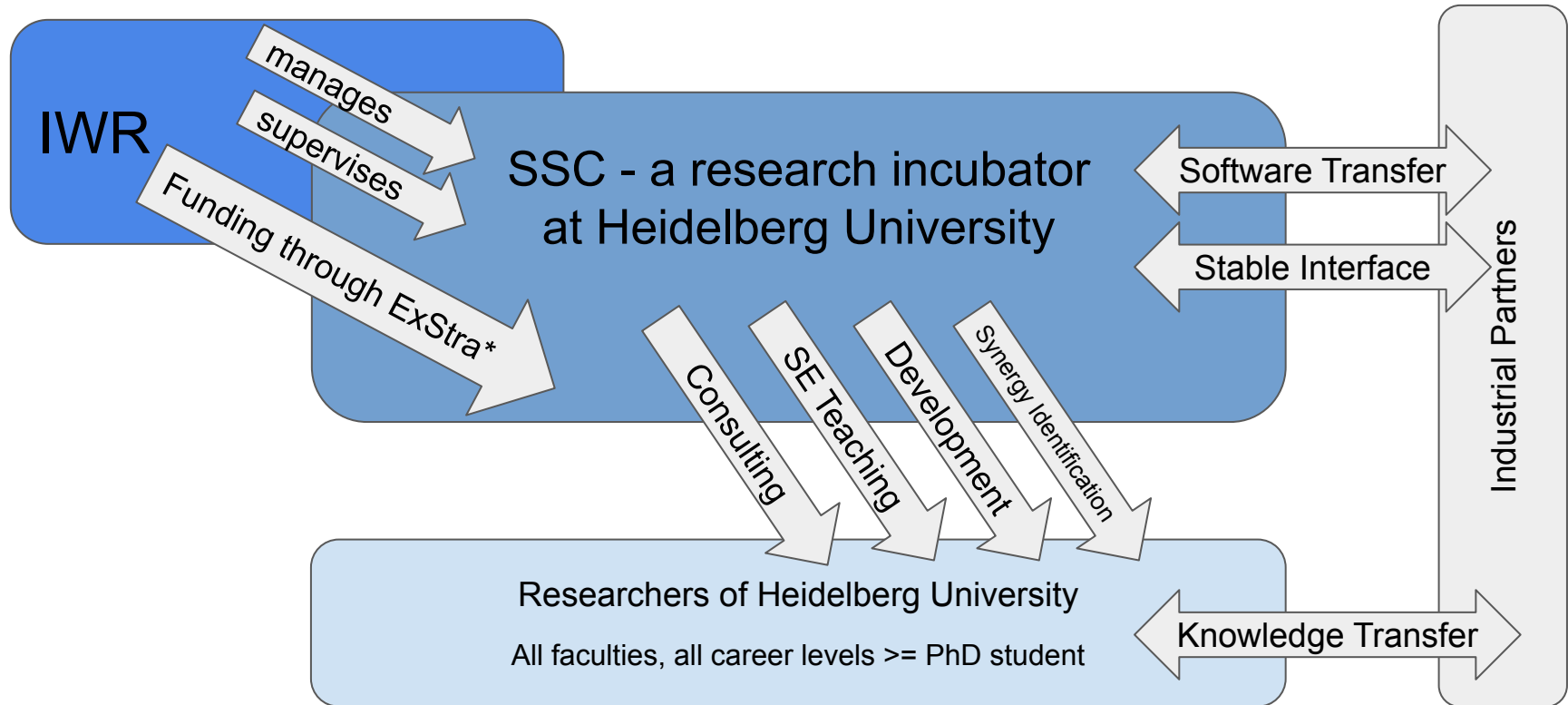
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

2024 Open Call of the Scientific Software Center

Liam Keegan, Dominic Kempf, Inga Ulusoy



The Scientific Software Center



*the SSC receives its core funding through the *Exzellenzuniversität* initiative

The SSC's target group

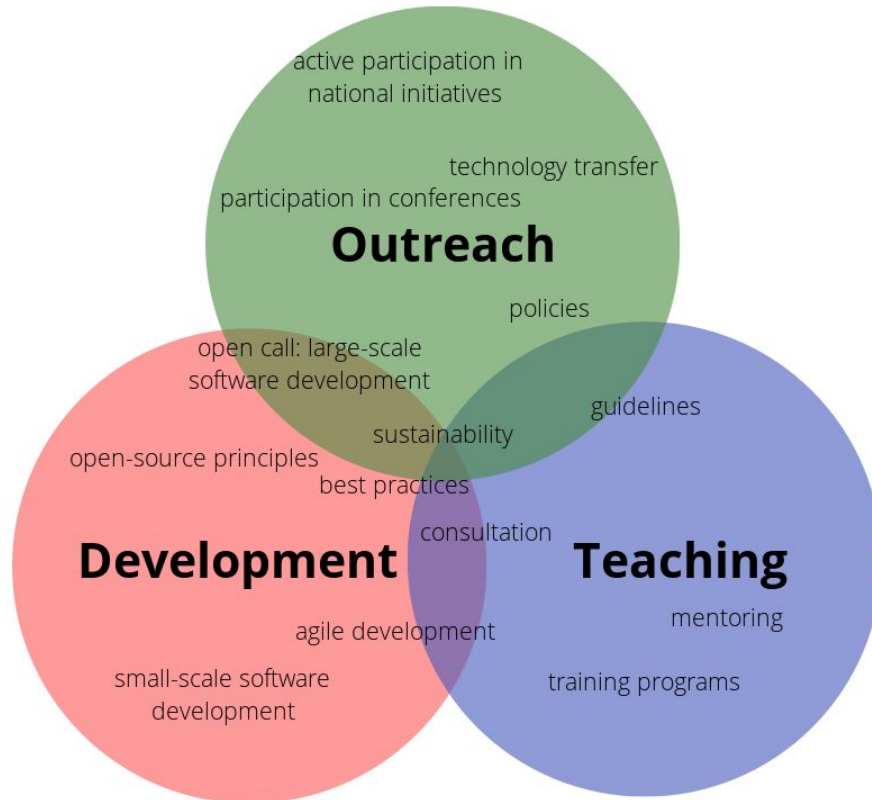
Users of specialized software/libraries

- Research critically depends on digital tools and specialized research software
- Adaptation of software to specific needs
- The generated data is used to provide scientific insight

Developers of own specialized software/libraries

- Software development and method development are core concepts that the research is based on
- The developed methodology and generated data are used to provide scientific insight

SSC mission and team



SSC Team

- Currently six Research Software Engineers (RSEs)
- Shared leadership:
 - Liam Keegan
 - Dominic Kempf
 - Inga Ulusoy
- System Administrator
- Administrative Support

Previous Open Call projects

2023:

- Improve performance of a Python photonic and acoustic simulation toolkit
- Refactoring a legacy fortran physics simulation codebase
- Best practices and software development guidance for a new astrophysics project
- Refactor, extend and add a GUI to a pipeline that quantifies cell morphology from electron microscopy
- Similarity search as a web service for identifying cultural heritage monuments from Nepal

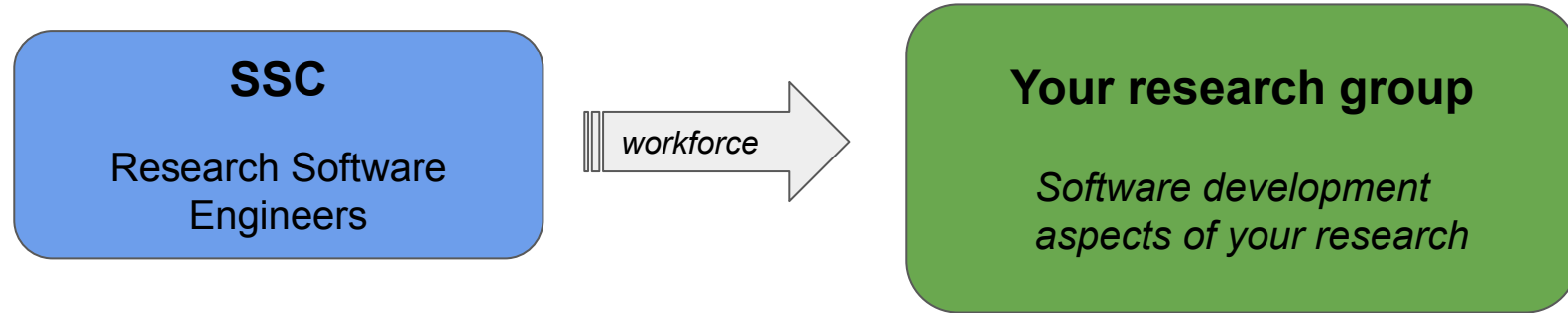
2022:

- Image analysis
- Text analysis
- Training of NLP models
- Generation of experiments in motor skill acquisition
- Modernize existing research software and apply SE best practices
- Improve maintainability and portability of existing research software

2021:

- Automated text annotation
- Neuroscience data processing
- GUI development for existing research software
- High-performance C++ implementation of existing Python code
- Python interface to research software including data transformations and GUI components

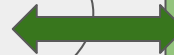
What can you apply for in this call?



- Custom development of **new scientific software** for a research project
- Addition of **new functionality** into existing research software
- Development that **increases the software quality** of existing research software (e.g. performance, scalability, portability, usability, reusability etc.)
- Adoption of **best practices** for the development of scientific software

Example 1: Custom development of new scientific software

Great research idea:
Expand into new methodological area



Methodology/concept has been developed on paper or in other research group; proof of concept

Your research group

- students
- Postdocs

knowledge transfer

Implementation of the methodology using efficient algorithms and modern libraries

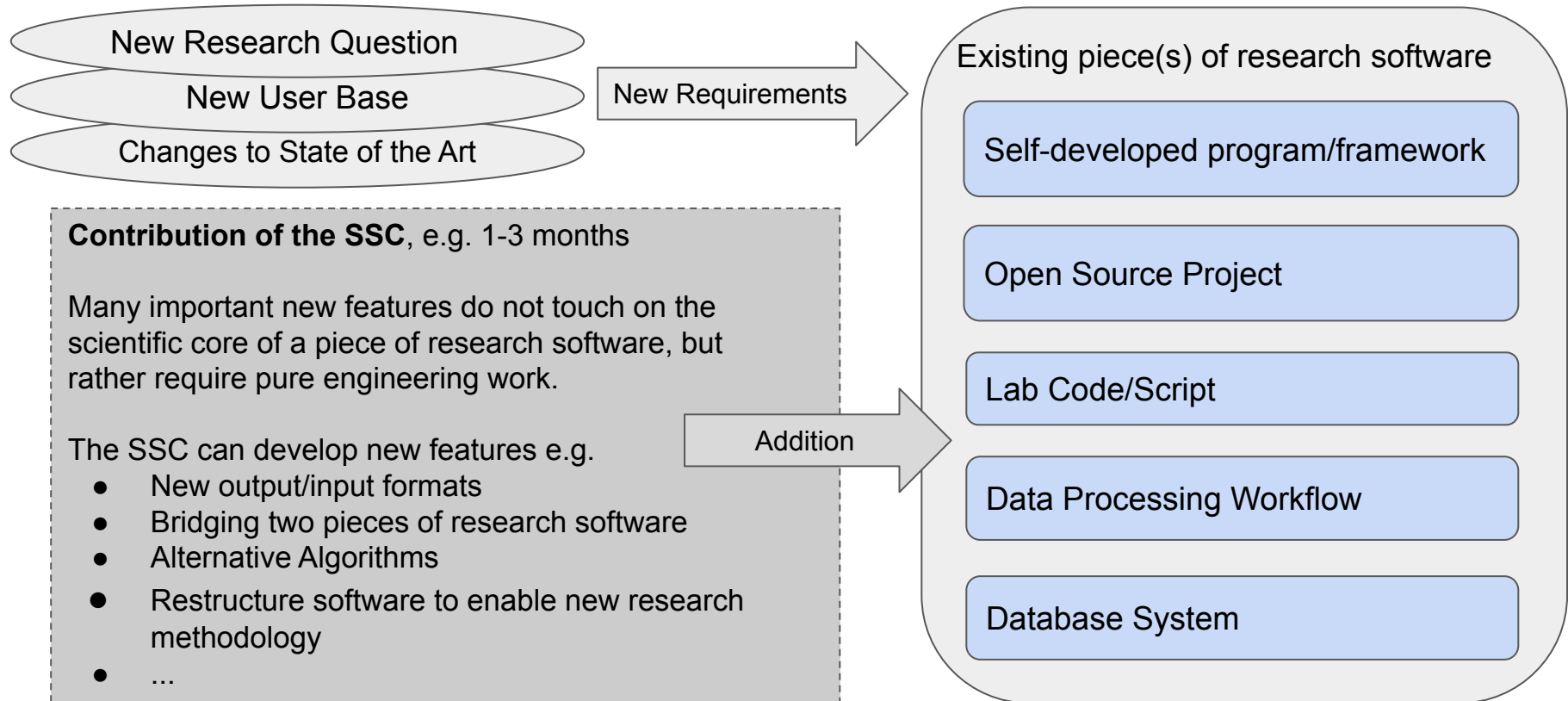
Test suite that ensures reproducibility; set up continuous integration

Documentation of the Software using modern tools and platforms; tutorials/examples

Version control and licensing models

Contribution of the SSC
e.g. 6 months of workforce

Example 2: New functionality



Example 3: Increase quality of existing research software

Involvement of students/postdoc knowledge transfer

Existing research software that has been developed in your group

- Low performance
- Low parallel scaling
- Dependence on a specific set of libraries (possibly outdated)

Contribution of the SSC

e.g. 3 months of workforce

Analysis and improvement of serial performance

Analysis and improvement of parallel performance

Suggestion and implementation of interface to alternative libraries; updated build system

Other examples could involve:

- portability of the software (different environments/operating systems)
- usability (user interface, code structure)
- reusability (generalization of the software)
- ...

Example 4: Adoption of best practices

Status Quo: A research group develops a lab code that is passed on from PhD generation to generation. Everybody adds functionality and examples from their work.

Code divergence

Insufficient Testing

“Grown” software design

Lack of Documentation

Introducing Git + GitHub/GitLab

Setting up CI + Introducing testing frameworks

Code refactoring: Software Design consultation + Sprint supervision

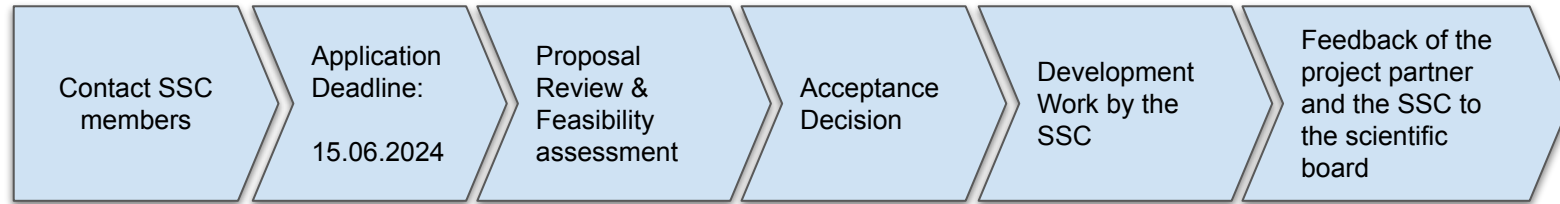
Introducing documentation tools
E.g. Sphinx + Doxygen

SSC Contribution: A developer joins the group and introduces best practices that follow the state of the art in software development.

Implementation: We set up tools and configurations for the group

Training: We teach the group to enable long term improvement

How to apply



- Applications can be submitted online:

www.ssc.uni-heidelberg.de/en/development/the-sscs-open-call

<https://limesurvey.urz.uni-heidelberg.de/index.php/189384>

- Applications can be written in English or German.

Evaluation criteria and process

The decision is taken by the scientific board of the SSC.

Criteria:

- Feasibility statement of the SSC
- Scientific merit of the proposal
- Clarity of what is expected from the SSC
- Leverage of project outcome
- Criteria about applicant:
 - Enabling interdisciplinary collaborations
 - Supporting Early Career Researchers
 - Aiding equal opportunities

Question Time!

Get in touch!

By email:

- ssc@uni-heidelberg.de

Or have a closer look at our services on our website:

- <https://ssc.uni-heidelberg.de>