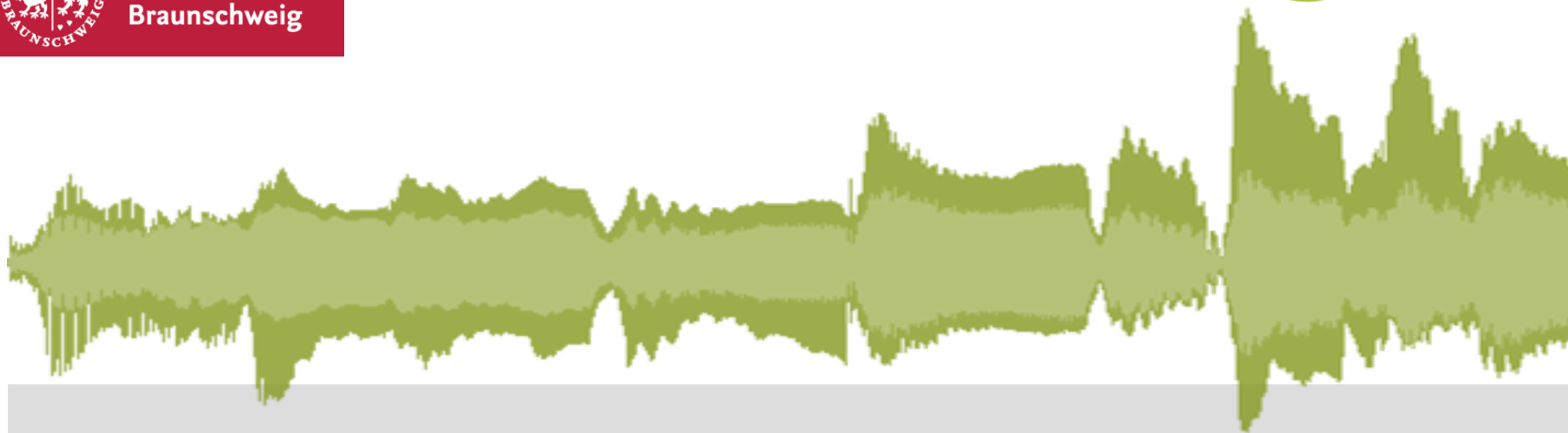




Technische
Universität
Braunschweig



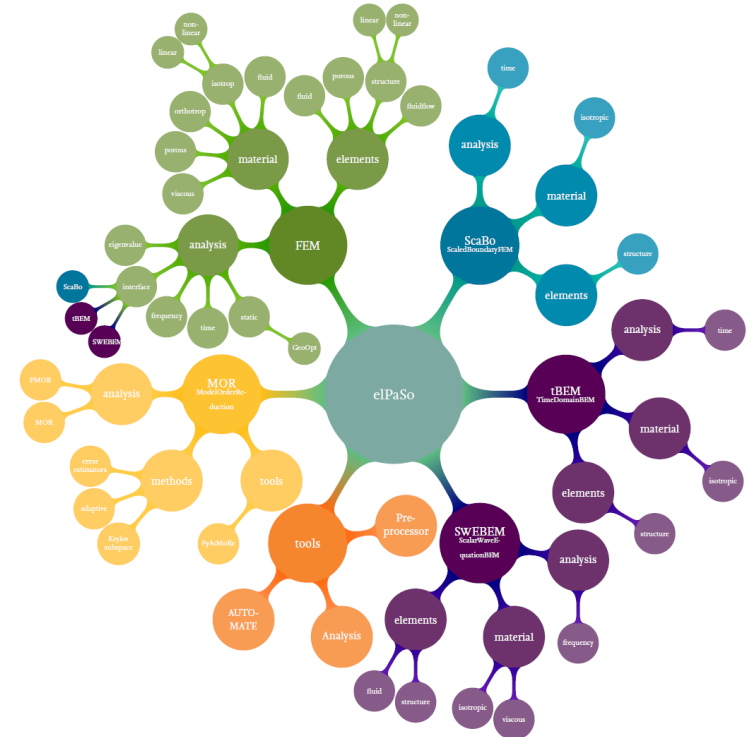
eIPaSo - In-House Research Code, Institute for Acoustics

Harikrishnan Sreekumar, 16 May 2022

eIPaSo | About

Elementary Parallel Solver (eIPaSo)

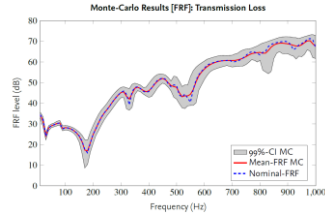
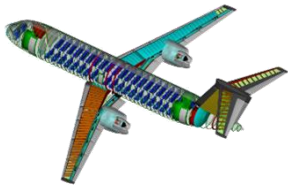
- Performs vibroacoustic analysis in the modal, static, time and frequency domain
- Based on FEM, BEM, SBFEM
- Efficient computing strategies - parallel computing, model order reduction



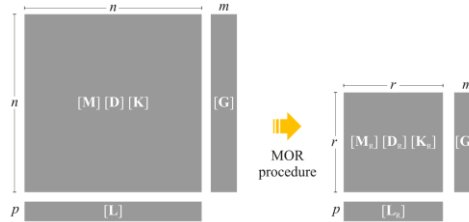
Source: InA/TU Braunschweig

eIPaSo | Key projects

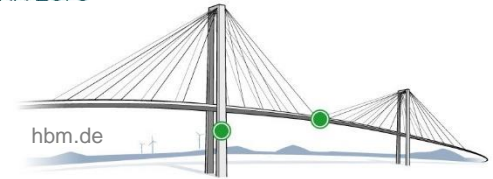
Aircraft cabin noise computations



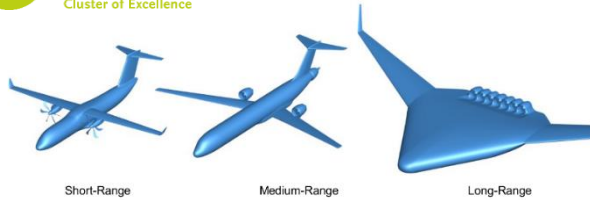
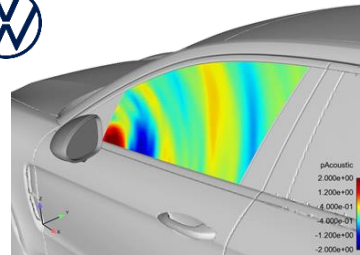
Model order reduction



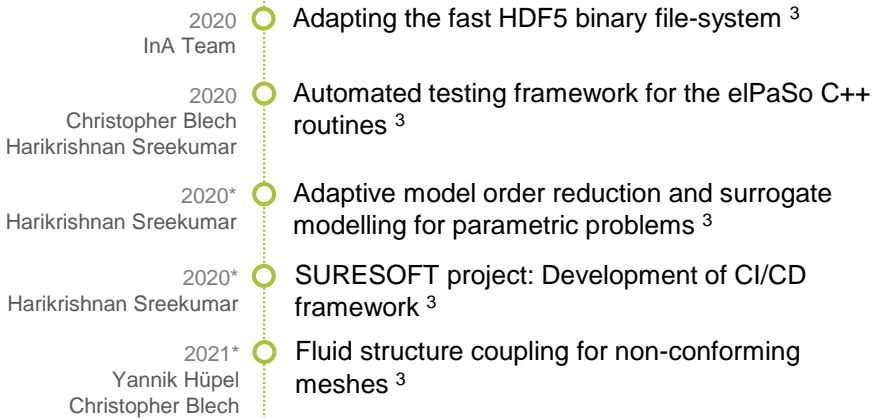
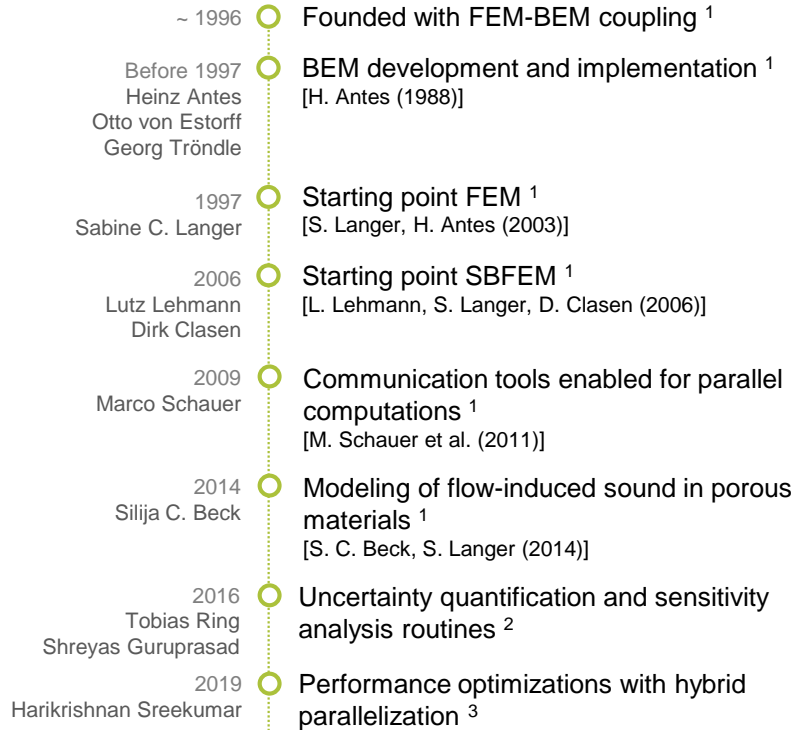
Data-driven structural health monitoring



Sound transmission computations



eIPaSo | Development timeline



* Active developments

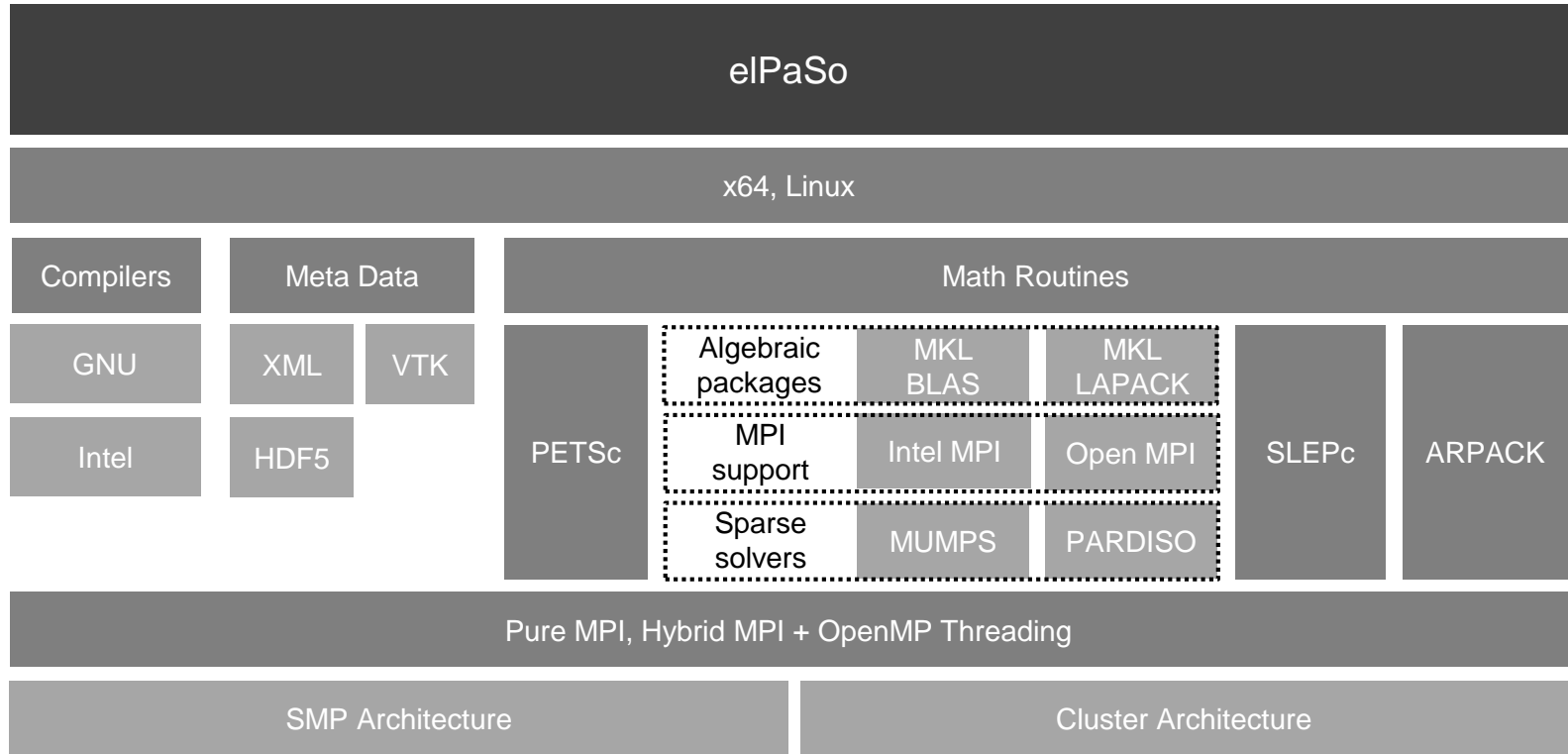
Other key authors: Lutz Ackermann, Katherina Rurkowska, Meike Wulkau, Sebastian Rothe, Mathias Dorn

¹ Institut für Angewandte Mechanik, TU Braunschweig

² Institut für Konstruktionstechnik, TU Braunschweig

³ Institut für Akustik, TU Braunschweig

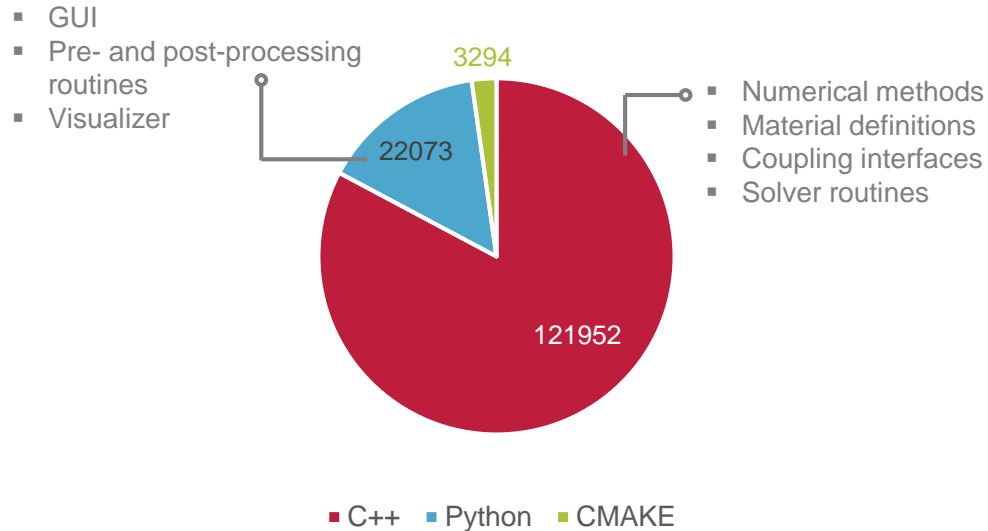
eIPaSo | Software architecture



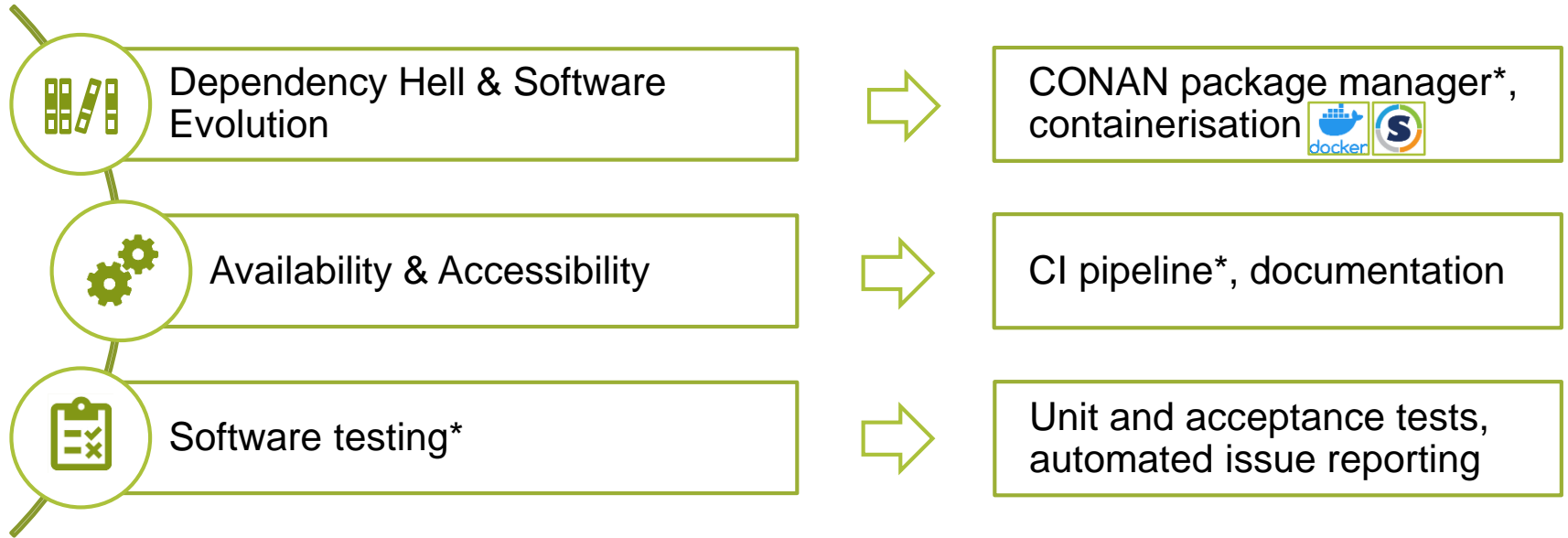
Source: InA/TU Braunschweig

eIPaSo | Source code

Programming language and SLOC:



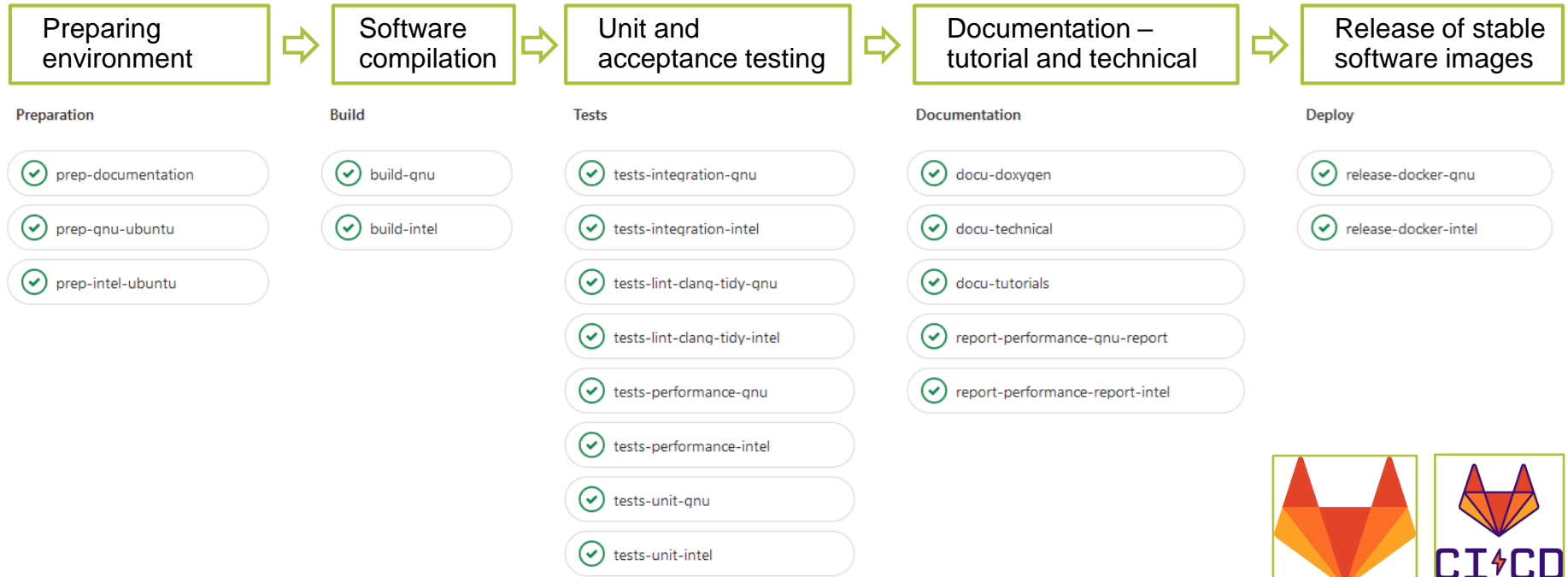
eIPaSo | Main challenges before SURESOFT



* details in coming slides

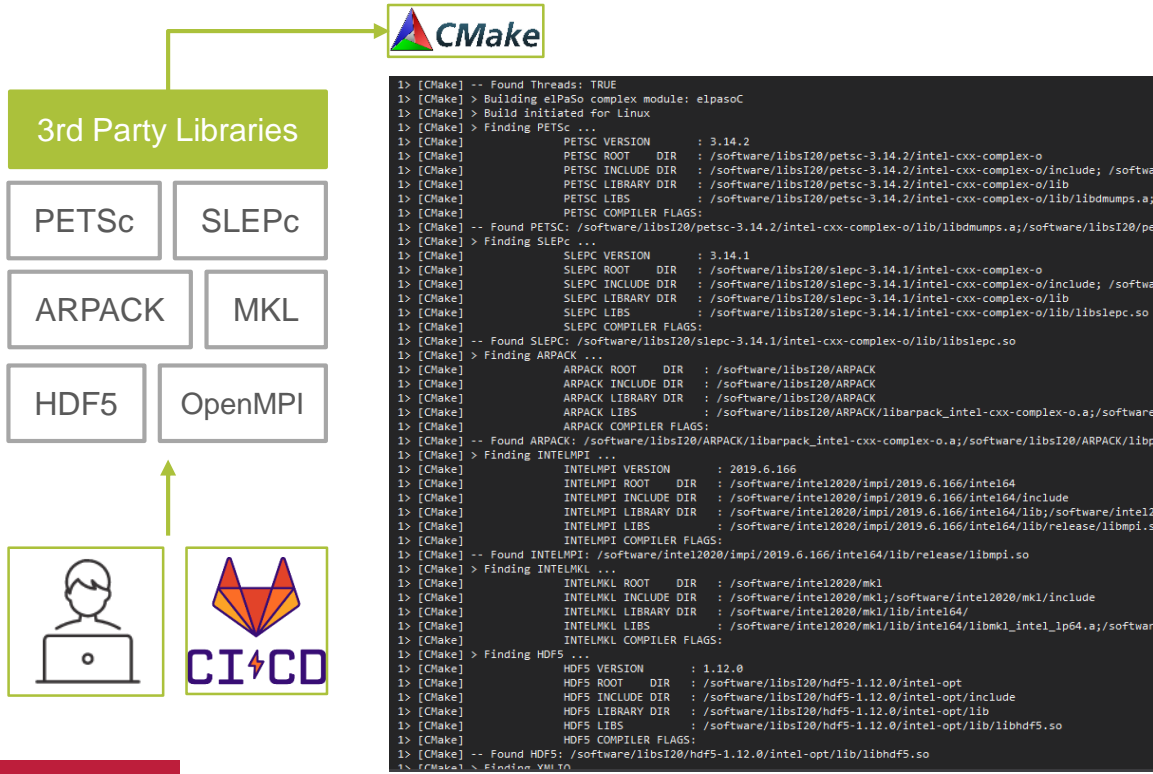
→ Need to be “ready to use” and “ready to develop”

eIPaSo | CI Pipeline



<https://git.rz.tu-bs.de/akustik/eIPaSo>

eIPaSo | Managing dependencies with CONAN.io



Without package manager:

- [] Library building time is minimum 2 hours
- [] Highly prone to configuration errors
- [] High coding knowledge required

eIPaSo | Managing dependencies with CONAN.io



CONAN registry in GitLab



```
1> [CMake] -- Downloading conan.cmake from https://github.com/conan-io/cmake-conan
1> [CMake] > Conan installation for intel compilers...
1> [CMake] -- Conan: checking conan executabe
1> [CMake] -- Conan: Found program /usr/bin/conan
1> [CMake] -- Conan: Version found Conan version 1.39.0
1> [CMake] |
1> [CMake] -- Conan executing: /usr/bin/conan install . --build missing --settings build_type=Release --s
1> [CMake] Configuration:
1> [CMake] [settings]
1> [CMake] arch=x86_64
1> [CMake] arch_build=x86_64
1> [CMake] build_type=Release
1> [CMake] compiler=intel
1> [CMake] compiler.libcxx=libstdc++11
1> [CMake] compiler.version=19.1
1> [CMake] os=Linux
1> [CMake] os_build=Linux
1> [CMake] [options]
1> [CMake] [build_requires]
1> [CMake] [env]
1> [CMake] conanfile.txt: Installing package
1> [CMake] Requirements
1> [CMake] arpack-real/2.1@ina+elpaso/stable from 'gitlab' - Cache
1> [CMake] hdf5/1.12.0@ina+elpaso/stable from 'gitlab' - Cache
1> [CMake] petsc-real/3.14.4@ina+elpaso/stable from 'gitlab' - Cache
1> [CMake] slepc-real/3.14.2@ina+elpaso/stable from 'gitlab' - Cache
1> [CMake] Packages
1> [CMake] arpack-real/2.1@ina+elpaso/stable:b60d3ded6fd63ebad90ada3d287cdfc4752f66d8 - Cache
1> [CMake] hdf5/1.12.0@ina+elpaso/stable:b60d3ded6fd63ebad90ada3d287cdfc4752f66d8 - Cache
1> [CMake] petsc-real/3.14.4@ina+elpaso/stable:b60d3ded6fd63ebad90ada3d287cdfc4752f66d8 - Cache
1> [CMake] slepc-real/3.14.2@ina+elpaso/stable:b60d3ded6fd63ebad90ada3d287cdfc4752f66d8 - Cache
1> [CMake]
1> [CMake] Installing (downloading, building) binaries...
1> [CMake] arpack-real/2.1@ina+elpaso/stable: Already installed!
1> [CMake] hdf5/1.12.0@ina+elpaso/stable: Already installed!
1> [CMake] petsc-real/3.14.4@ina+elpaso/stable: Already installed!
1> [CMake] slepc-real/3.14.2@ina+elpaso/stable: Already installed!
1> [CMake] conanfile.txt: Generator txt created conanbuildinfo.txt
1> [CMake] conanfile.txt: Generator cmake_find_package created Findpetsc-real.cmake
1> [CMake] conanfile.txt: Generator cmake_find_package created Findslepc-real.cmake
1> [CMake] conanfile.txt: Generator cmake_find_package created Findarpack-real.cmake
1> [CMake] conanfile.txt: Generator cmake_find_package created Findhdf5.cmake
1> [CMake] conanfile.txt: Aggregating env generators
1> [CMake] conanfile.txt: Generated conaninfo.txt
```

openmpi/4.0.0@ina+elpaso/stable
4.0.0 Conan

hdf5/1.12.0@ina+elpaso/stable
1.12.0 Conan

arpack-real/2.1@ina+elpaso/stable
2.1 Conan

arpack-complex/2.1@ina+elpaso/stable
2.1 Conan

slepc-real/3.14.2@ina+elpaso/stable
3.14.2 Conan

slepc-complex/3.14.2@ina+elpaso/stable
3.14.2 Conan

petsc-complex/3.14.4@ina+elpaso/stable
3.14.4 Conan

petsc-real/3.14.4@ina+elpaso/stable
3.14.4 Conan

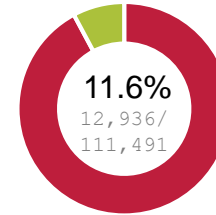
<https://git.rz.tu-bs.de/akustik/eIPaSo/-/packages>



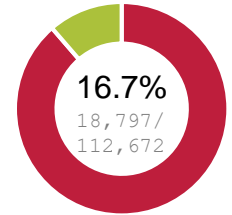
eIPaSo | Software testing

Unit testing with GoogleTest [C++]

- New codes → Test driven development
- Legacy codes → Refactoring and testing



Unit test coverage



Acceptance test coverage

Acceptance testing with in-house tool AUTOMATE (Automated testing tool for eIPaSo)

- Predefined set of vibroacoustic benchmark problems
- Correctness tests – Compares solution with the benchmarked using an error measure
- Performance tests – Compares execution time accounted from various timers

Features of AUTOMATE tool

- Issue reporting – `python-gitlab` for automated issue creation in GITLAB issue board
- Detailed technical report (currently generated as PDF, in future also as Gitlab pages)

eIPaSo | Future developments

- Scientific **documentation** into the development lifecycle
- Software **licensing** – Open source licensed core functionalities and closed source licensed new research implementation
- Making eIPaSo **citable**
- Benchmark models, results and containers to be **archived**
- Enhancing **code quality**

Thank you for your attention

Harikrishnan Sreekumar
Institute for Acoustics
Technische Universität Braunschweig
hk.sreekumar@tu-braunschweig.de