Call to Action: Integrate with Karabo

SKACH Winter Meeting

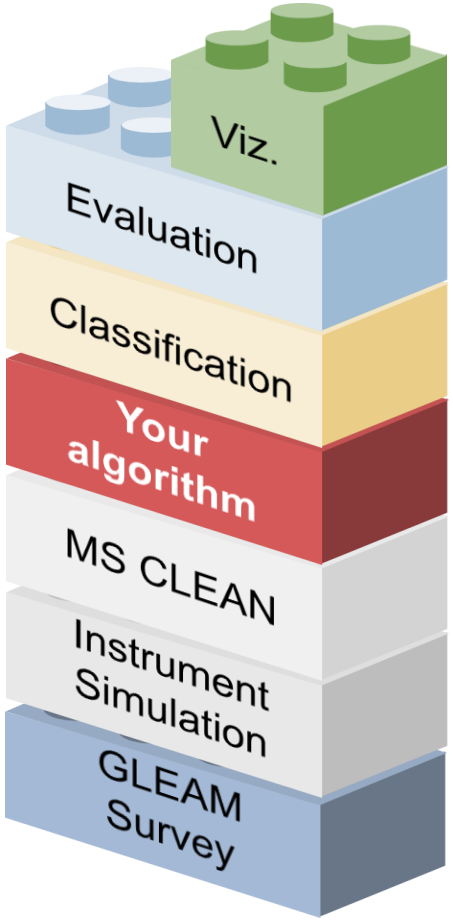
2023-01-12

Simon Felix (simon.felix@fhnw.ch)
Vincenzo Timmel (Vincenzo.timmel@fhnw.ch)

SKACH

ETH zürich

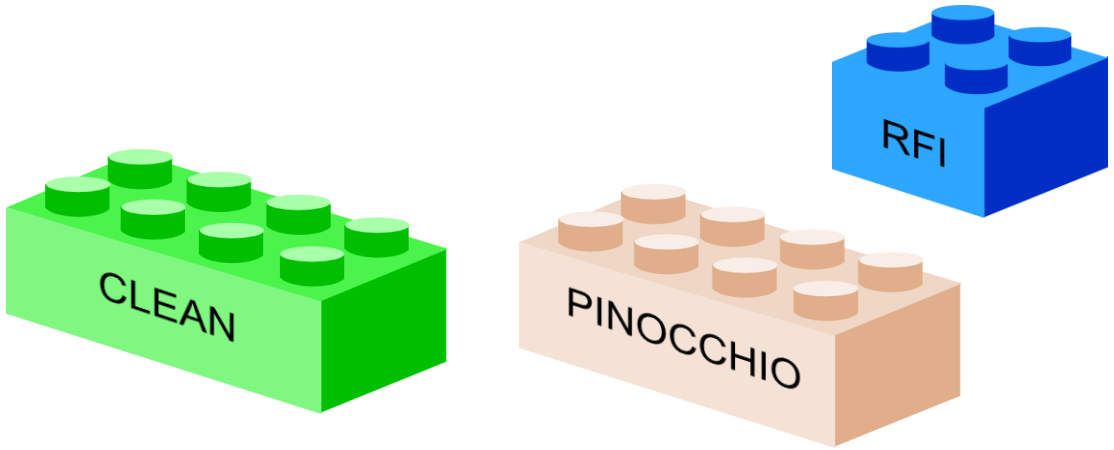
University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering



Fast and easy ramp-up.

Build **custom** pipelines with our building blocks.

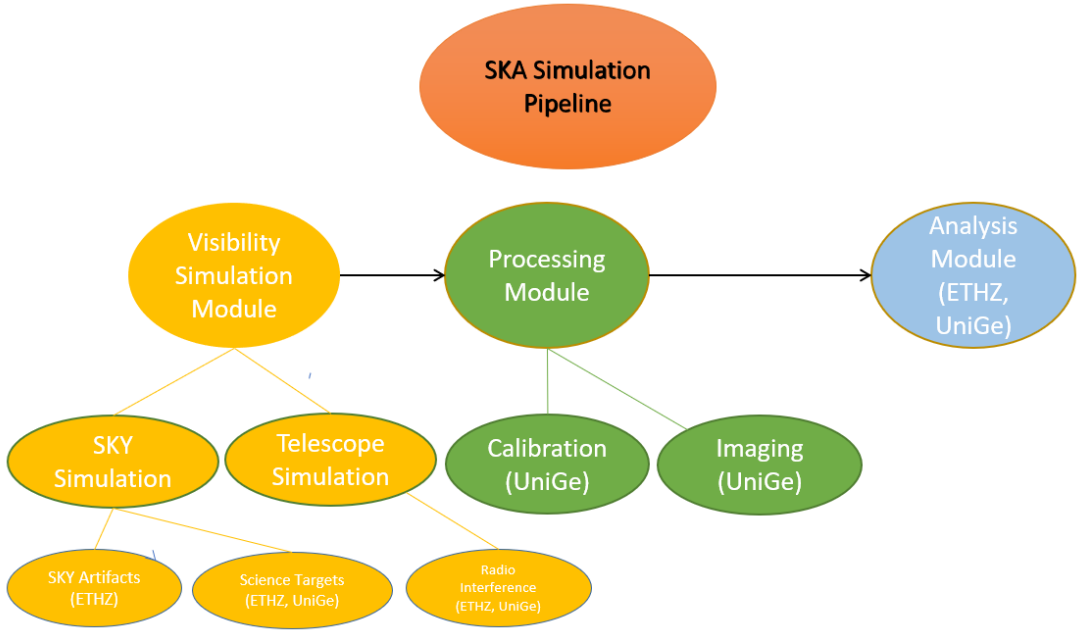
Add your own blocks.

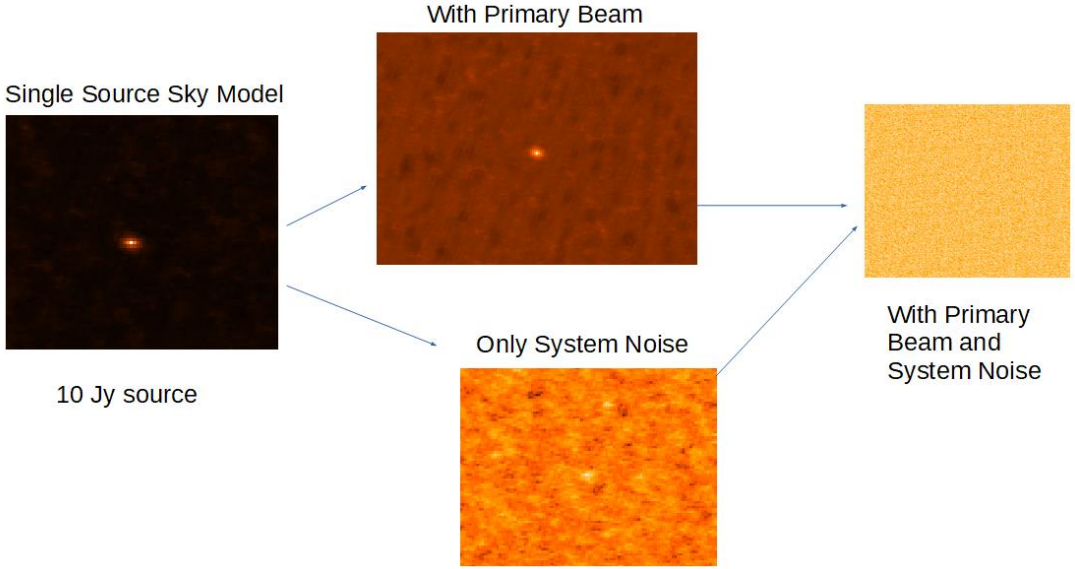


State of Karabo

Features

- Custom sky models
- Catalogs (e.g. GLEAM, MIGHTEE)
- Telescope configurations
- Point and shoot observations
- Custom primary beams
- Noise simulation
- Source detection
- Dark matter halo simulation





Integrated Software

- OSKAR
- RASCIL
- PINOCCHIO
- PyBDSF
- Sklearn
- EIDOS
- KatBeam
- WSCLEAN (in progress)

Runtime Environments

Locally
Jupyter Notebooks
...

Containers

HPC Environments

Binaries for x64 Linux

Installation

Conda Local

```
conda create -n karabo-env
 python=3.9
conda activate karabo-env
conda install -c i4ds -c conda-
 forge -c nvidia/label/cuda-11.7.0
 karabo-pipeline
```

Docker Jupyter

```
docker pull ghcr.io/i4ds/karabo-
 pipeline:jupyter
docker run -p 8888:8888 -v
 ska_pipeline_code:/home/jovyan/wo
 rk/persistent
 ghcr.io/i4ds/karabo-
 pipeline:jupyter

# run this at the top of your nb
from karabo.util.jupyter import
set_rascil_data_directory_env
set_rascil_data_directory_env()
```

Docker CLI

```
docker pull ghcr.io/i4ds/karabo-
 pipeline:cli
docker run -it
 ghcr.io/i4ds/karabo-pipeline:cli
```

Roadmap

2022

- Installation and operation
- Source detection workflow
- Initial datasets
- Common telescope configurations
- CI/CD setup

2023

- Improved stability
- Improved usability
- Tests on CSCS
- More Workloads and Algorithms (Bluebild, Innovative Imaging Algorithms, ...)
- More datasets
- **Your Code?**

Continuous Integration and Deployment (CI/CD)

- Karabo components are **automatically** and **regularly** tested
- We release Karabo once per month (https://github.com/i4Ds/Karabo-Pipeline/releases)
- Documentation is generated and published automatically (https://i4ds.github.io/Karabo-Pipeline/)

We want your software too!

- We build, test and package your code for you
- We ensure interoperability with other software for you
- We distribute your software and data for you

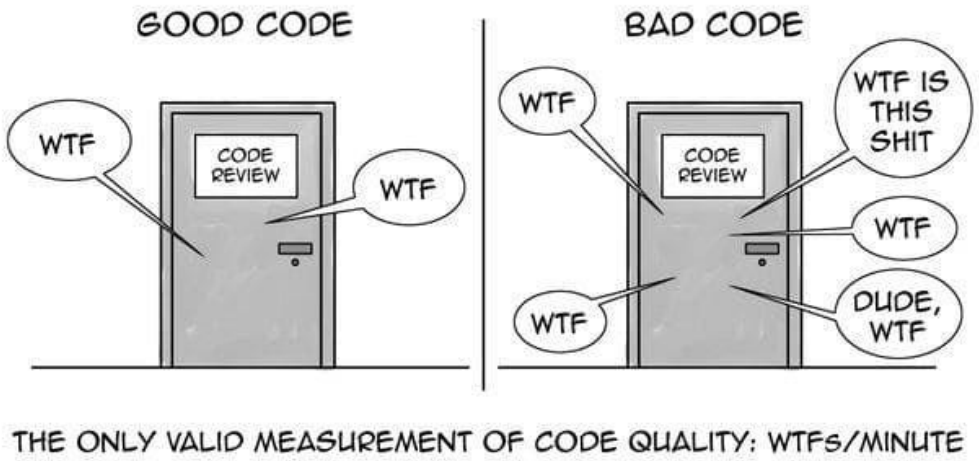
→ Your software is available to a broader audience (with or without Karabo)

→ Your software is easier to install for others

→ Your software is tested and distributed long-term

What can you add?

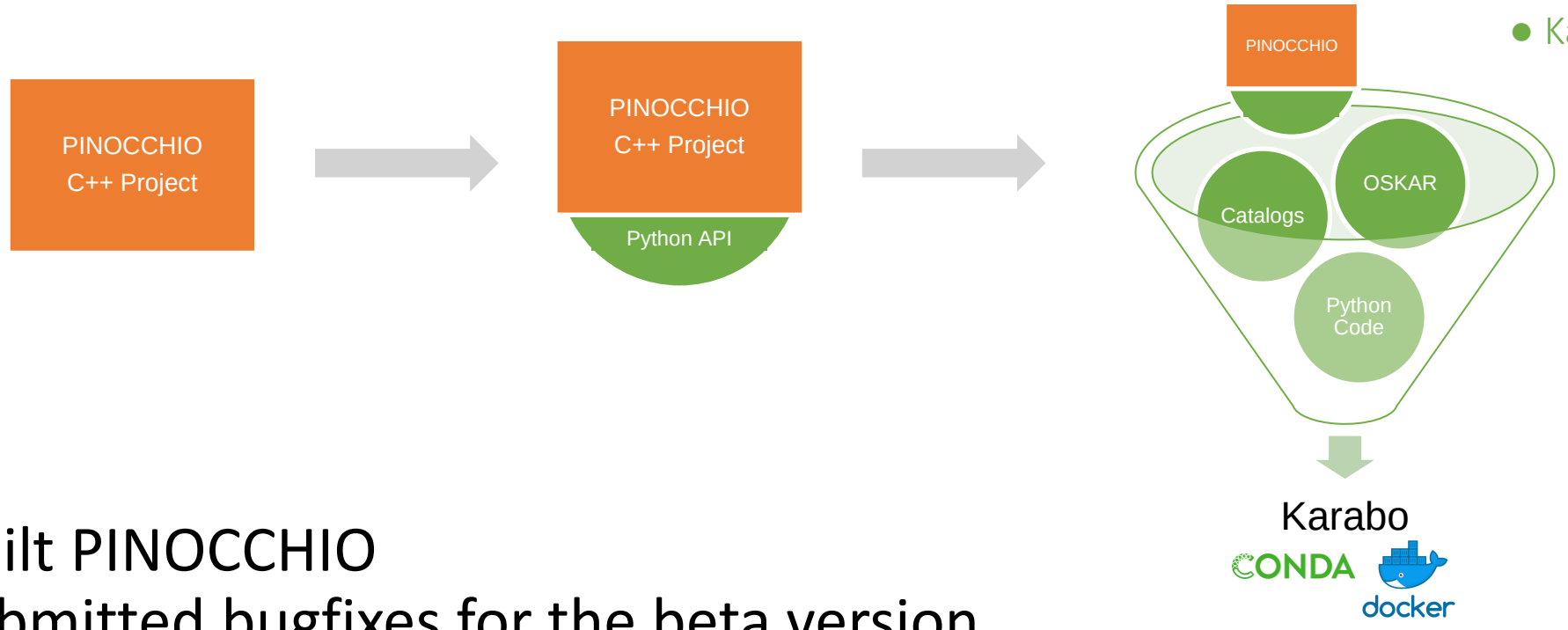
- Code in Python, C++, Fortran, ... which depends on CUDA, MPI, …
- Imaging Algorithms, Source Detection, Quick Look Generation, Simulations, Catalogs, Machine Learning Models, Visualizations, Analyses, Code Snippets, Examples, Sky Models, Benchmarks, RFI models, Workflows, …



- Useful, but unpolished and unfinished research code

Example: Adding PINOCCHIO to Karabo

- Pierluigi Monaco
- Karabo Team



1. We built PINOCCHIO
   We submitted bugfixes for the beta version
2. We published PINOCCHIO publicly with Conda
3. We bundle Conda packages to produce Karabo

What will we write?

We add a Karabo-compatible Python API around your code.

We test basic functionality, but no details.

If you’re interested, we’ll help you add additional tests.

If you’re interested, we’ll help you write documentation.

What should you do?

Send us your code

Send us a link to your repository

Send us the name of your Conda package

…and we’ll take it from there.

Important: It’s still your code afterwards.

Important: We’re not asking to modify your software.

GitHub

http://github.com/i4ds/Karabo-Pipeline

Code & Issues

slack

https://swissska.slack.com/

Chat in #digital-twin

Report Issues

Request Features

Get Support

Contribute

Do you spend effort to build software and deployment?

Is there anything stopping you from testing our offer?
Are there worrying parts?

Do you prefer to manage everything yourself?
What parts would you be happy to collaborate on?