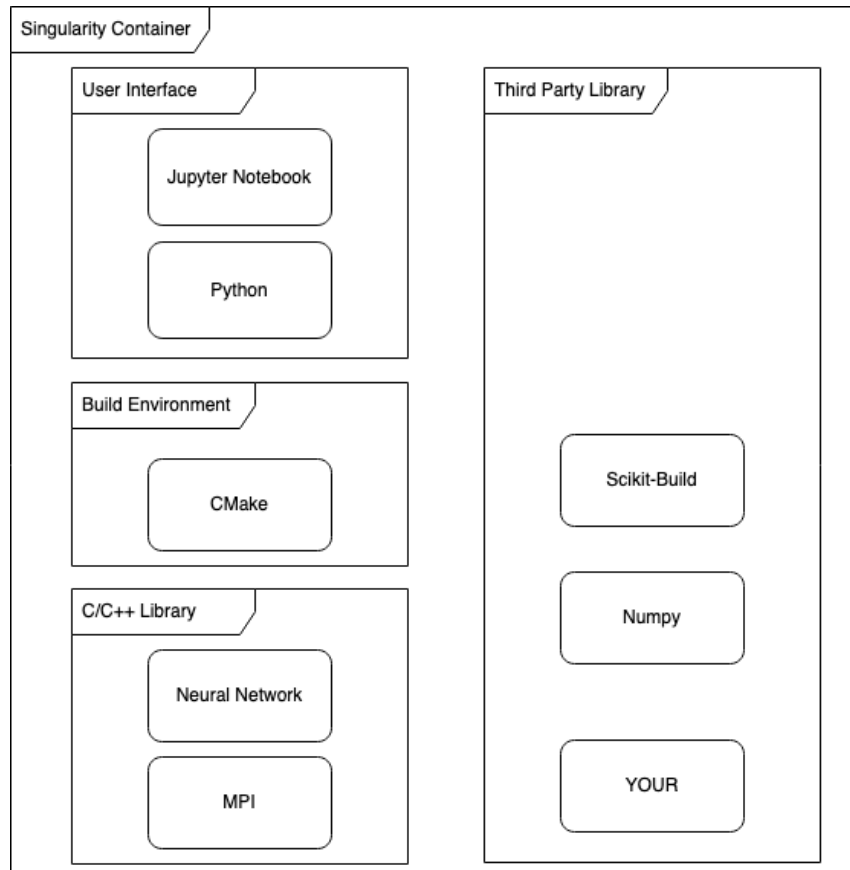


Marcel Trattner

# Pulsar Data Analysis

## Update on Tool Status

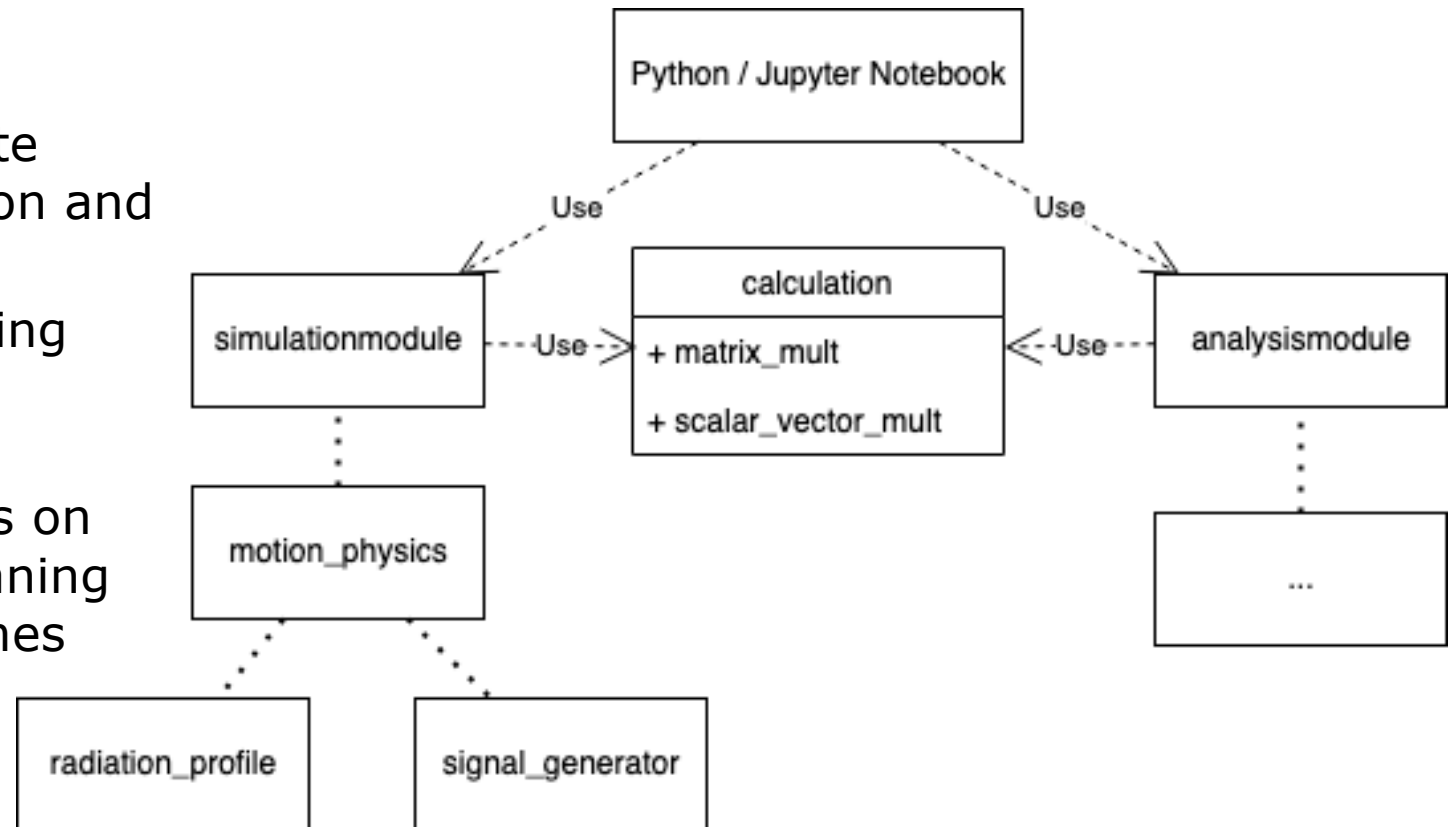
# Overall Architecture



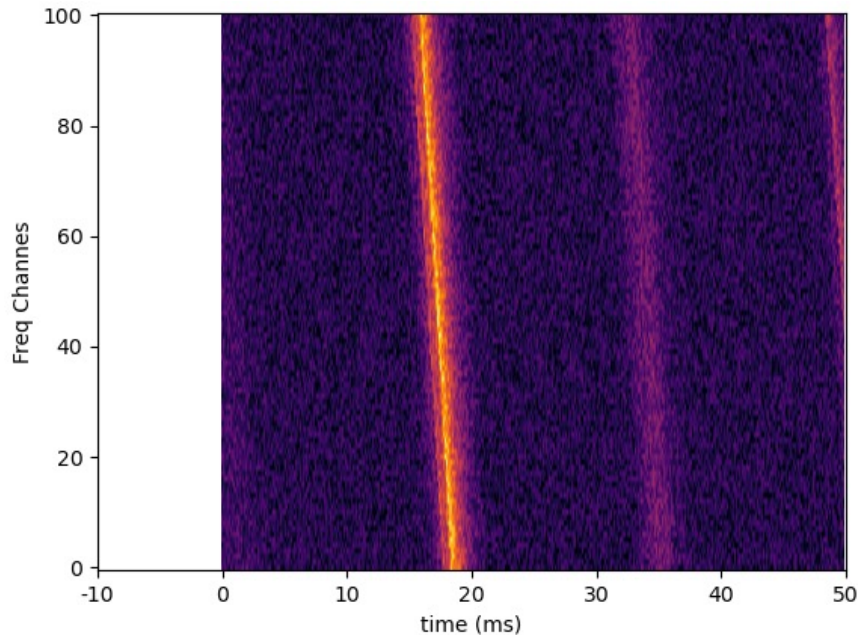
- Python can be extended with custom C/C++ library
- Performance of C is generally better and will be used for the time critical aspects
- CMake can build even complex projects and can be integrated into Python with the tool Scikit-Build

# Project Packaging Structure

- Creating two separate libraries for simulation and analysis
- Translating the existing python code for the simulation into C++
- Putting the emphasis on performance and planing parallelization schemes

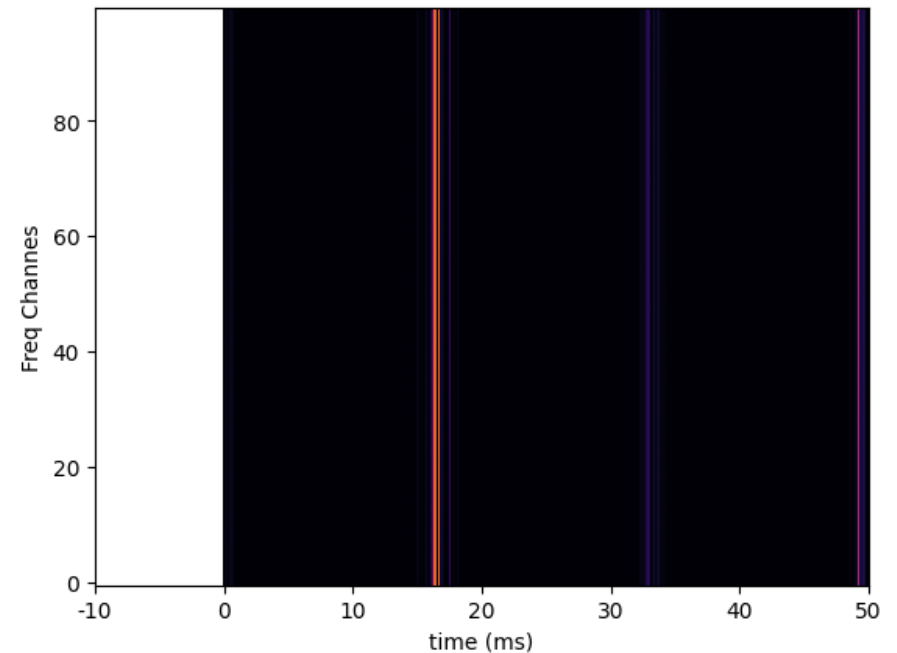


# Current state



Python simulation:  
generate\_frequency\_timeGraph  
V2

[Saha, 2023]



C++ translation:  
generate\_frequency\_timeGraph  
(V1)

[Trattner, 2023]

# Next steps

- Continuing implementation of simulation
- Following software engineering best practices to ensure maintainability and maximal performance
- Investigating into ideal solution for the analysis and differentiation of pulses and RFIs.
- Scaling the application across multiple nodes and running performance tests



**Hochschule für Technik  
und Wirtschaft Berlin**

University of Applied Sciences

[www.htw-berlin.de](http://www.htw-berlin.de)