

# Reliable digital twin simulation development and execution on HPC infrastructures

Marek Kasztelnik<sup>(1)</sup>, Tomasz Gubała<sup>(1,2)</sup>, Piotr Nowakowski<sup>(1,2)</sup>, Jan Meizner<sup>(1,2)</sup>, Piotr Połec<sup>(1)</sup>, Maciej Malawski<sup>(1,2)</sup>, Marian Bubak<sup>(1,2)</sup>  
<sup>(1)</sup> ACC Cyfronet AGH University of Science and Technology, Kraków, Poland  
<sup>(2)</sup> Sano Centre for Computational Medicine, Kraków, Poland

## Motivation

- Creating simulations and testing them is time-consuming
- HPC systems are not intuitive, introducing an additional layer of complexity
- The standard way of using HPC systems does not provide repeatability and reproducibility; it also lacks modern versioning (such as is provided by Git)
- It is difficult to monitor tasks
- Terminal access to HPC requires some knowledge

## Solution

The Model Execution Environment can run simulations on HPC from a web browser. It is integrated with Git repositories, where the simulation source code should be stored. The user has the ability to choose which branch/version to use; then MEE uploads and executes the code. During execution, status is monitored, and result files can be downloaded afterwards. Pipelines can have many steps that can be launched automatically when all required inputs are present, or executed in parallel if possible. MEE also takes care of the underlying file structure, making it easy to upload and download simulation data and results.

## Architecture

- Patient – virtual space to run different calculations on patient data.
- Flow – Blueprint for creating pipelines quickly – it represents a collection of steps (or instructions) that comprise the pipeline.
- Step – Single instruction or program. Used to assemble flows. Each step can specify what type of input is required to start the calculation.
- Pipeline – Simulation created for a given patient. Interfaces HPC and delivers output files. Can specify which version of code will be used. Can be either automatic or manual.
- Model – set of scripts/source code stored in a Git repository.

## Running simulations

- Check if all required inputs are present
- Fetch model starting script template from Git repository
- version specified by user
- Generate model starting script basing on inputs provided by the user
- Submit job to HPC queuing system (Slurm)
- Upload required inputs to HPC
- Monitor job execution
- Fetch results from HPC

## Sample use case

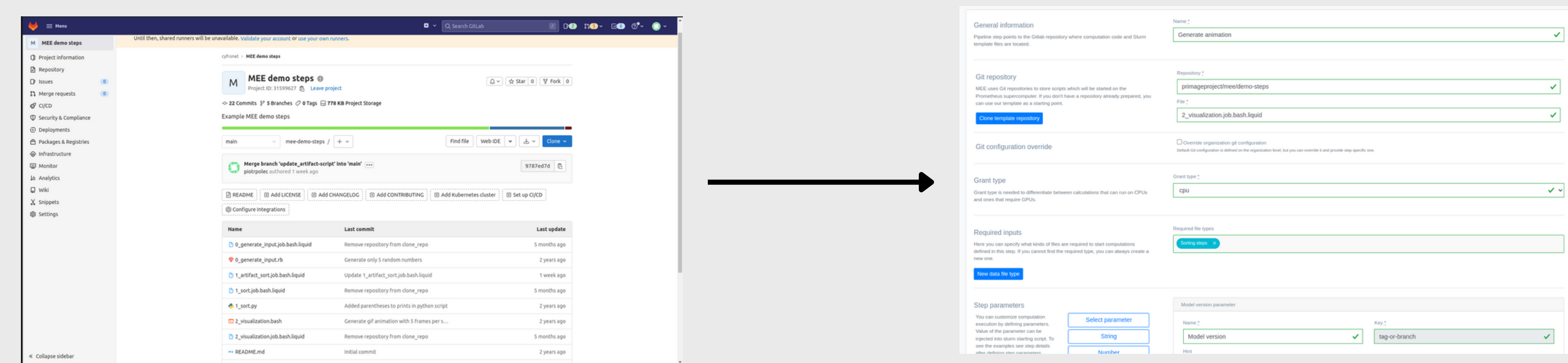


Fig.1 Git repository is used to store pipeline step code

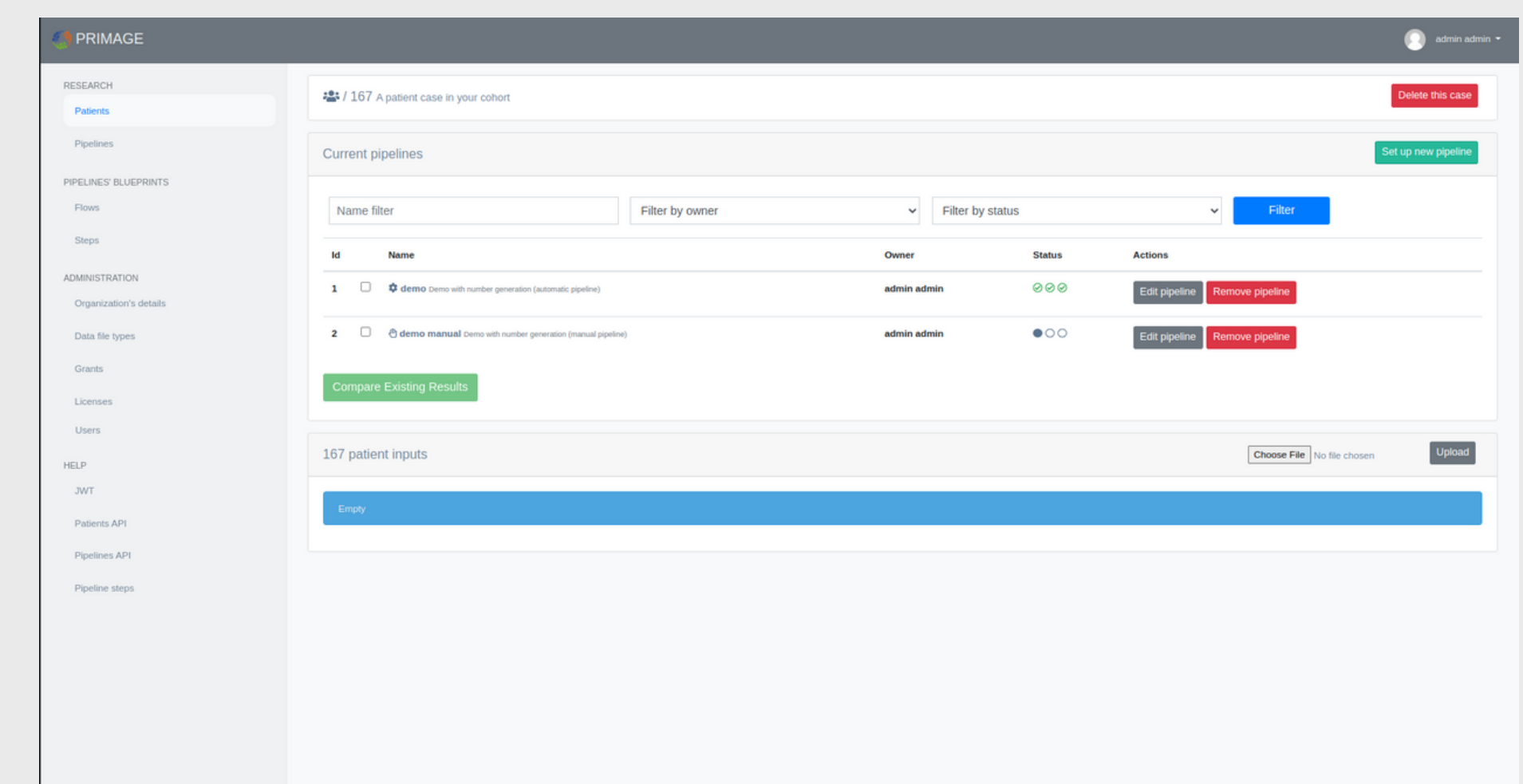


Fig.2 Patient view with pipelines and their current status (completed; available for execution; missing inputs)

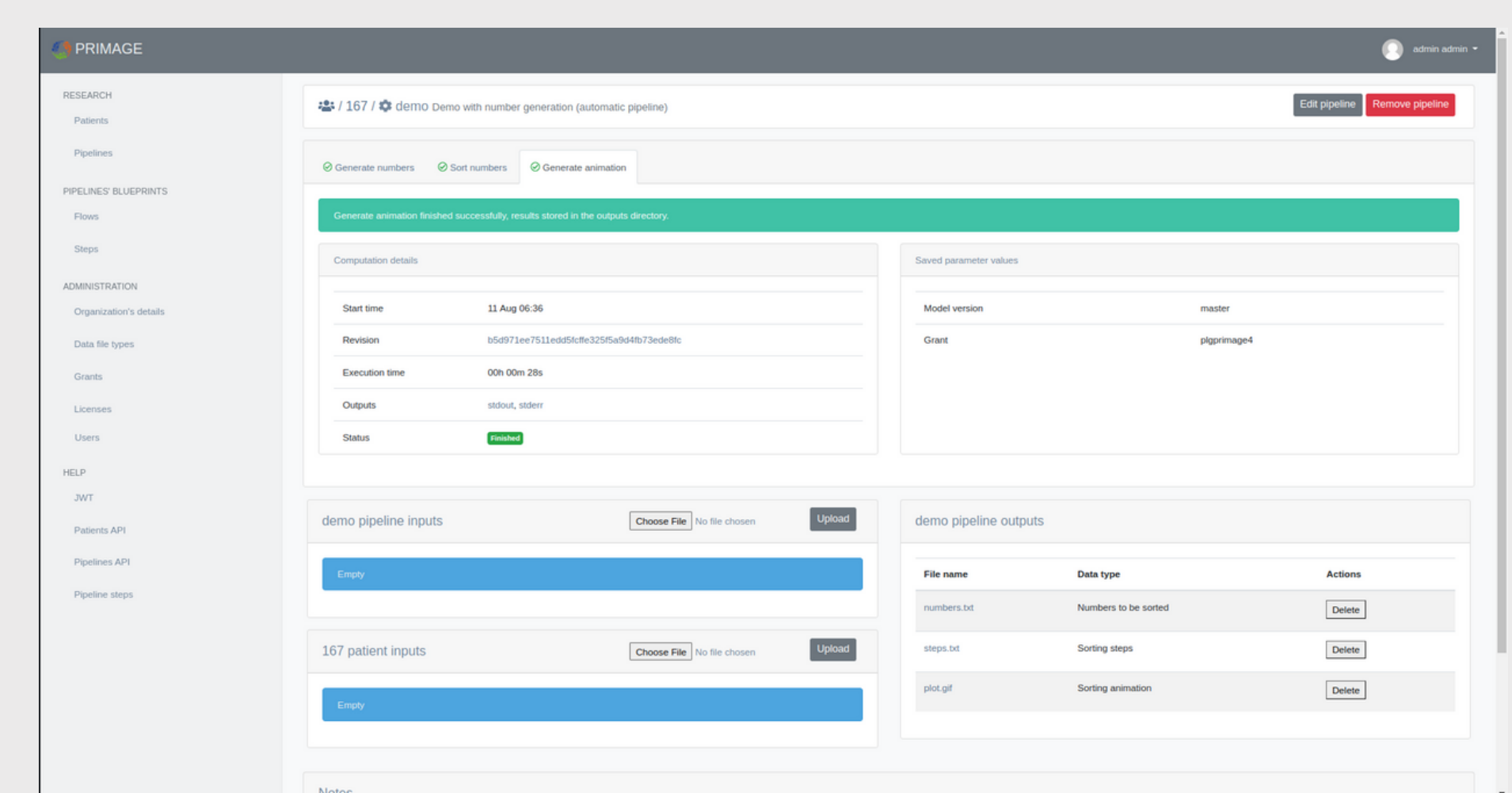
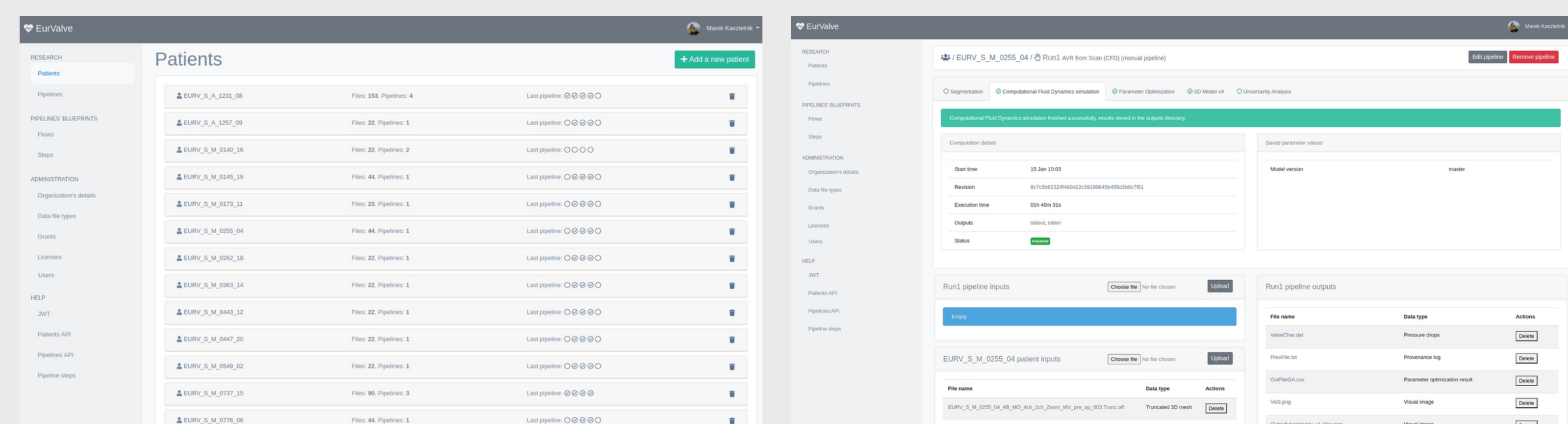


Fig.3 Pipeline view. All details about a specific simulation – its status and the version that was ran. The user can download output files.

## Uses so far

- Lung disease classification
- Differential expression profile analysis; variant search
- Forensic genotyping
- Computational fluid dynamics for cardiac simulations
- Cardiac uncertainty analysis
- Cardiac pressure volume loop comparison
- Cardiac reduced order model

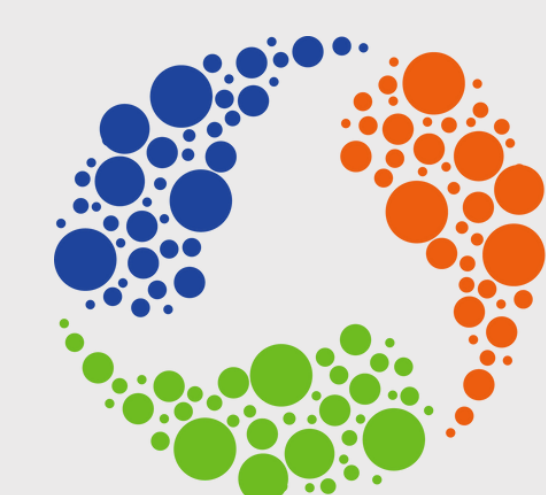


### References:

1. M Bubak, K Czechowicz, T Gubała, DR Hose, M Kasztelnik, M Malawski, J Meizner, P Nowakowski, S Wood The EurValve model execution environment, Interface Focus 11 (1), 20200006, 2021

### Acknowledgments:

This work was supported by the EU project PRIMAGE H2020-EU.3.1.5.3 826494 and by the PL-Grid Infrastructure ([www.plgrid.pl](http://www.plgrid.pl)).



**PRIMAGE**  
Medical imaging  
Artificial intelligence  
Childhood cancer research