interTwin logo


**D4.6 Final Architecture design of the DTs capabilities for High Energy Physics, Radio astronomy and Gravitational-wave Astrophysics**

**Status: Under EC Review**

**Dissemination Level: public**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Description | | | |
| **D4.6 Final Architecture design of the DTs capabilities for High Energy Physics, Radio astronomy and Gravitational-wave Astrophysics** | | | |
| **Work Package 4** | | | |
| **Document type** | Deliverable | | |
| **Document status** | UNDER EC REVIEW | **Version** | 1.0 |
| **Dissemination Level** | Public | | |
| **Copyright Status** | This material by Parties of the interTwin Consortium is licensed under a [**Creative Commons Attribution 4.0 International License**](http://creativecommons.org/licenses/by/4.0/). | | |
| **Lead Partner** | INFN | | |
| **Document link** | [**https://documents.egi.eu/document/3942**](https://documents.egi.eu/document/3942) | | |
| **DOI** | [**https://zenodo.org/records/15120028**](https://zenodo.org/records/15120028) | | |
| **Author(s)** | * Kalliopi Tsolaki (CERN) * Sofia Vallecorsa (CERN) * Sara Vallero (INFN) * Lorenzo Asprea (INFN) * Francesco Sarandrea (INFN) * Javad Komijani (ETHZ) * Gaurav Sinha Ray (CSIC) * Yurii Pidopryhora (MPG) * Isabel Campos (CSIC) | | |
| **Reviewers** | * Jorge Gomes (LIP) * Rakesh Sarma (FZJ) | | |
| **Moderated by:** | * Andrea Anzanello (EGI Foundation) | | |
| **Approved by** | * Isabel Campos (CSIC) on behalf of TCB | | |

|  |  |
| --- | --- |
| **Abstract** | |
| **Key Words** | Architecture, design, capabilities, Digital Twin Engine, High Energy Physics, GW Astrophysics, Radio Astronomy |
| This deliverable provides an updated report on the final architecture of Digital Twins (DTs) in the physics domain, highlighting their evolution since initial designs and detailing the technical rationale behind architectural choices. It incorporates user stories to address the diverse requirements of stakeholders, including DT operators and end users, ranging from domain experts to non-expert users. Additionally, the report outlines how DT Applications interact with the interTwin Digital Twin Engine (DTE) to process data, compose workflows, and visualize results, integrating modules from various Work Packages (WPs) within the project. | |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| **Version** | **Date** | **Description** | **Contributors** |
| V0.1 | 05/12/2024 | Template creation from the 1st architecture design | Andrea Cristofori (EGI) |
| V0.2 | 27/02/2025 | ToC updated and sections assigned | Francesco Sarandrea (INFN) |
| V0.3 | 07/03/2025 | All contribution added | All the authors |
| V0.4 | 07/03/2025 | Final check of contributions before internal review | Francesco Sarandrea (INFN) |
| V0.5 | 14/03/2025 | Internal review | Jorge Gomes (LIP), Rakesh Sarma (FZJ) |
| V0.6 | 21/03/2025 | Correction requested by reviewers | All the authors |
| V0.7 | 31/03/2025 | TCB review | Isabel Campos (CSIC) |
| v0.8 | 31/03/2025 | Version ready for QA | Francesco Sarandrea (INFN) |
| **V1.0** |  | **Final** |  |

|  |  |
| --- | --- |
| Terminology / Acronyms | |
| **Term/Acronym** | **Definition** |
| AI | Artificial Intelligence |
| ANNALISA | Advanced Nonlinear transient-Noise Analyser of Laser Interferometer Sensor Arrays |
| ACS | Access Control Service |
| CNN | Convolutional Neural Network |
| DT | Digital Twin |
| DTE | Digital Twin Engine |
| FC | File Catalogue |
| GAN | Generative Adversarial Network |
| GNN | Generative Neural Network |
| GW | Gravitational Wave |
| HDF5 | Hierarchical Data Format version 5 |
| HL-LHC | High Luminosity - Large Hadron Collider |
| HEP | High Energy Physics |
| HPC | High Performance Computing |
| HPO | Hyper Parameter Optimization |
| IAM | Identity and Access Management |
| IdP | Identity Provider |
| ILDG | International Lattice Data Grid |
| MC | Monte Carlo |
| MCMC | Markov Chain Monte Carlo |
| MDC | Metadata Catalogue |
| ML | Machine Learning |
| ML-PPA | Machine Learning-based Pipeline for Pulsar Analysis |
| NF | Normalizing Flow |
| NN | Neural Network |
| ONNX | Open Neural Network Exchange |
| QCD | Quantum Chromodynamics |
| SE | Storage Element |
| TB | Terabyte |
| WP | Work Package |

Terminology / Acronyms: [**https://confluence.egi.eu/display/EGIG**](https://confluence.egi.eu/display/EGIG)

Table of Contents

[1 Introduction 9](#_Toc194414311)

[1.1 Aim of this deliverable 9](#_Toc194414312)

[1.2 Intended audience of this document 9](#_Toc194414313)

[1.3 Structure of the document 9](#_Toc194414314)

[2 DT Applications User Stories 10](#_Toc194414315)

[2.1 DT Application: Lattice QCD simulation 10](#_Toc194414316)

[2.2 DT Application: Detector simulation 12](#_Toc194414317)

[2.3 DT Application: Noise simulation for radio astronomy 14](#_Toc194414318)

[2.4 DT Application: VIRGO Noise Detector 16](#_Toc194414319)

[3 DT Applications Design 18](#_Toc194414320)

[3.1 DT Application: Lattice QCD simulation 18](#_Toc194414321)

[3.1.1 Advanced data management for Lattice QCD 18](#_Toc194414322)

[3.1.2 Generative models using Machine Learning 20](#_Toc194414323)

[3.2 DT Application: Detector simulation 22](#_Toc194414324)

[3.3 DT Application: Noise simulation for radio astronomy 25](#_Toc194414325)

[3.4 DT Application: VIRGO Noise detector 28](#_Toc194414326)

[4 Conclusions 31](#_Toc194414327)

[5 References 32](#_Toc194414328)

List of Figures

[Figure 1 - Module Integration Diagram for the Lattice QCD use case 10](#_Toc194412473)

[Figure 2 - Schema depicting the flow of control when accessing the Lattice Datalake 19](#_Toc194412474)

[Figure 3 - Graphical representation of the classical generation of configurations using MC algorithms and of the Normalizing Flows method 21](#_Toc194412475)

[Figure 4 – Fast particle detector simulation using ML techniques high level workflow composition 23](#_Toc194412476)

[Figure 5 - Detailed graph representation of the training and inference workflows composition of the fast particle detector simulation DT 25](#_Toc194412477)

[Figure 6 - General outline of the DT structure 26](#_Toc194412478)

[Figure 7 - Diagram of the ML-PPA (in the C4 model) 26](#_Toc194412479)

[Figure 8 - Layered software architecture of the framework ML–PPA 27](#_Toc194412480)

[Figure 9 - The Virgo DT schema 28](#_Toc194412481)

[Figure 10 - System Context diagram of the DT for the veto pipeline 30](#_Toc194412482)

List of Tables

[Table 1 – User stories for DT Application: Lattice QCD Application 11](#_Toc194412683)

[Table 2 – User stories for DT Application: Detector Simulation 13](#_Toc194412684)

[Table 3 – User stories for DT Application: Noise Simulation for Radio Astronomy 15](#_Toc194412685)

[Table 4 – User stories for DT Application: VIRGO Noise Detector 16](#_Toc194412686)

**Executive summary**

This document is Deliverable 4.6 (D4.6) of the interTwin project, part of Work Package (WP) 4. It is a report collectively written by the partners of tasks 4.1, 4.2, 4.3 and 4.4, who are directly involved in designing digital twins for the physics domain (High Energy Physics (HEP), Radio Astronomy, and Gravitational Wave Astrophysics). This report is an update of previous Deliverable 4.4 (D4.4) and it highlights the final architecture of the Digital Twins (DTs) and how they have evolved since their first designs. This document illustrates the progress made during the project and details the technical motivations for the specific choices in the different architectures.

[**Section 2**](#_DT_Applications_User) briefly describes the DTs and the projects' main stakeholders. [**Section 3**](#_DT_Applications_Design) contains a more detailed description of their designs and integration with the DTEs.

We outline here the final design choices for each DT. The Lattice Quantum Chromodynamics (QCD) DT employs the normflow package for the generation of lattice configurations, giving the user the ability to sample from different physical theories. The Detector Simulation DT comprises a Geant4-based simulation framework and the deep learning GAN component, and both systems are integrated with the itwinai component as Kubeflow containers. The core of the Radio Astronomy DT consists of a CNN trained with both real and simulated astronomical data; once again, the modules are conveniently containerized. Lastly, the DT for the simulation of noise in Gravitational Wave signals consists of a training and an inference subsystem, composed of a collection of Kubernetes pods orchestrated by AirFlow.

# Introduction

## Aim of this deliverable

The overall objective of D4.6 is to provide an overview of the final versions of the DT Applications, their features, and their architecture design for the physics domain (T4.1, T4.2, T4.3, T4.4) and their key requirements in the interTwin project. A **Digital Twin Application** is a user interface implementation of a DT. DT applications are the consumers of the capabilities offered by the interTwin Digital Twin Engine (DTE), therefore they introduce use case-specific requirements.

## Intended audience of this document

The main audiences for D4.6 are the **developers** and **end users**. For the DT application and DTE **developers**: This deliverable is a succinct summary of the specifications and details about different components, data integration strategies, and computational models exploited to build the functioning DTs. This document serves both as a record of the state of the project in its final stage and as a reference point from which to progress and make new final features and improvements; such improvements can, for example, enhance the scalability of the models, leverage components and workflows, better technical support over time and better interoperability. In particular, developers of the DTE can use the document to construct a mapping between DTE components and DT applications and certify that the requirements from the DT Applications are satisfied at this stage, while developers of the DT Applications can use this document as a reference for their own requirements and for the way they plan to finalise the integration with the underlying DTE modules.

For DT **end users** and **operators**, this deliverable facilitates data sharing, integration, and analysis among various stakeholders. By establishing a common framework for communication, stakeholders will be able to exchange information, validate models, and collaboratively address their challenges. The document will use the term **DT User** to refer to the intended users and the main stakeholders for each DT application.

## Structure of the document

The structure of this deliverable is as follows. [**Section 2**](#_DT_Applications_User) describes the user interface and requirements for each digital twin. A detailed table is provided for each digital twin application where details are provided regarding user stories, their requirements (following the MoSCoW method [[**R1**](#_References)]), expectations and timeframe for completing the tasks. [**Section 3**](#_DT_Applications_Design) explains the architecture design and illustrates the workflow composition within each DT application. It depicts sequential or parallel steps involved in operating each DT, highlighting the input, processing, and expected outcomes.

# DT Applications User Stories

## DT Application: Lattice QCD simulation

Task 4.1 is developing a DT application for the simulation of quantum field theories on lattices. Two scenarios are being explored; a conventional scenario for Lattice QCD simulations, where large scale simulations take place on High Performance Computing (HPC) systems, and a second scenario based on Machine Learning (ML) accelerated simulations. The primary stakeholder is the DT developer who designs ML architectures for learning quantum field theories, trains models using these architectures, and then evaluates the trained models. The other stakeholder is the DT user who exploits these trained models for their research by using them to generate lattice configurations over which they can then measure physical observables. Figure 3 of Deliverable 7.8 illustrates these complementary workflows and is reproduced here as **Figure 1** for ease of reference [[**R2**](#_References)].

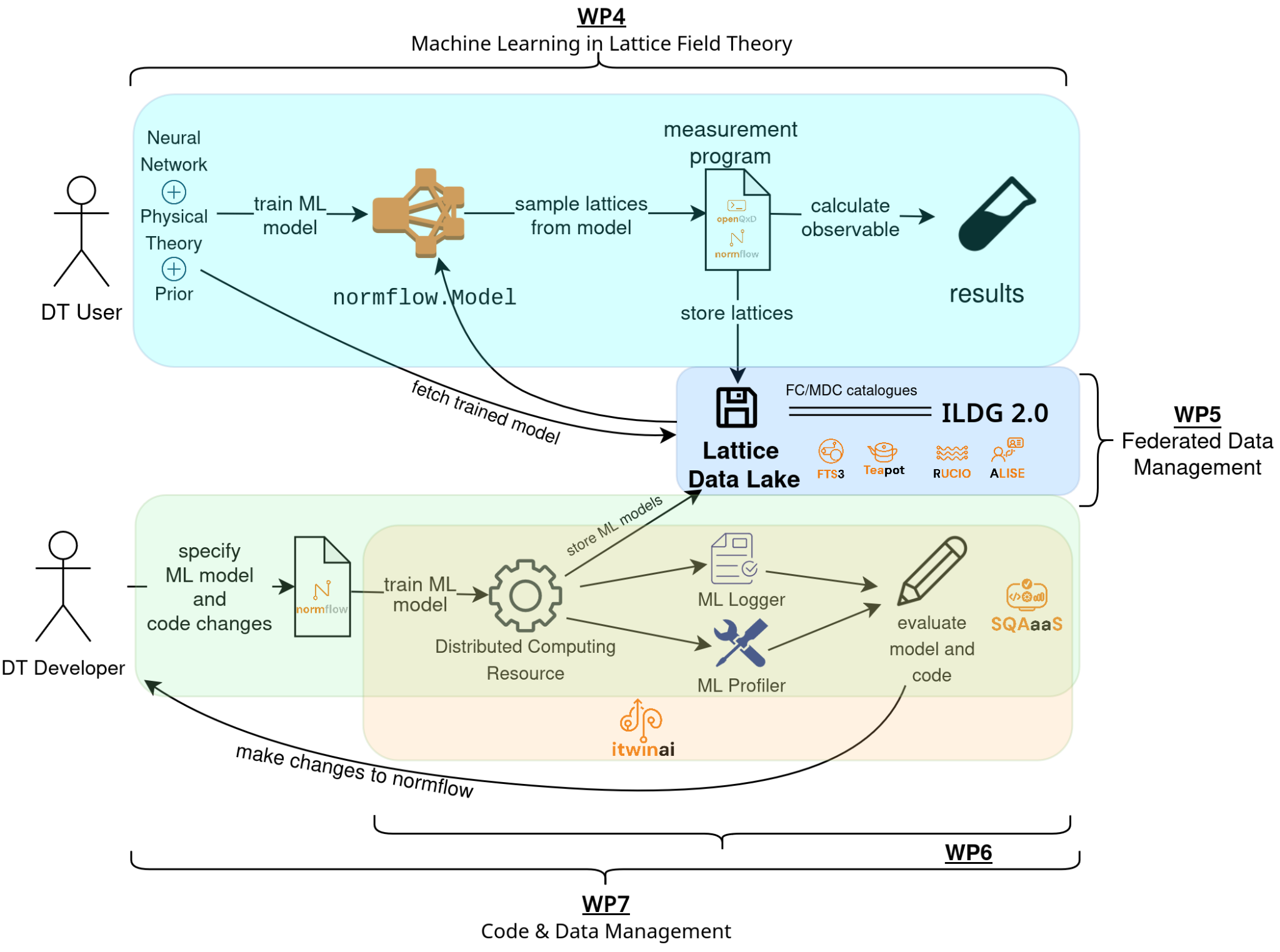


Figure 1 - Module Integration Diagram for the Lattice QCD use case

Large scale Markov Chain Monte Carlo (MCMC) simulations of Lattice QCD produce large amounts of valuable data that researchers would like to store for later re-use. Increasing redundancy and enabling fast data transfer is fundamentally important as current competitive simulations require transfers of O(100)TB between HPC centres. Currently lattice researchers are subject to the policies of their respective HPC centres. At best this unnecessarily hinders them from sharing data if one of them has not registered a classic ssh-based account on the corresponding system, at worst it outright prevents them from sharing their data.

Task 4.1 has been exploring the benefits to a typical lattice collaboration of the federated data capabilities, in particular the so-called “datalakes”, enabled by the tools developed within the interTwin project [[**R3**](#_References)]. The datalake framework allows one to use the International Lattice Data Grid (ILDG) [[**R4**](#_References)] as an Identity Provider (IdP). Instead of needing to register with multiple HPC centres, a lattice researcher only needs to be registered with the ILDG to access a “Lattice Datalake” that contains data uploaded by members of the lattice community to many different HPC centres. Using ILDG’s Oauth2 token-based authorisation also affords administrators more control over per-user permissions, this is vital given the size and international nature of lattice collaborations. Using the ILDG as an IdP should also ease future integration with the ILDG’s File (FC) and Metadata (MDC) catalogues, which are expected to grow to be the premier repositories for raw lattice data in the coming years.

An active area of research is the study of whether and how ML techniques like Normalising Flows (NF) can complement and augment conventional MCMC lattice simulations [[**R5**](#_References)]. Through the development of the *normflow* [[**R6**](#_References)] package, it has been shown that NF can be used for lattice field configuration generation with scalar theories [[[**R**](https://docs.google.com/document/d/17cu4vImdJHxk-sw41nlC1NtW2UUv7Qk1/edit#heading=h.2dlolyb)**7**](#_References)]. The *normflow* package has additionally been developed to handle the more complicated family of SU(N) gauge theories [[**R8**](#_References)]. This is an important step towards the ultimate goal of an NF-based lattice simulation of QCD, since QCD is based on the SU(3) gauge theory. The models trained using *normflow* can then be employed in a pipeline to calculate physical observables of interest, e.g. the magnetisation in scalar 𝜙⁴ theory. By generating lattice configurations relatively cheaply, one may study the behaviour of observables more precisely, or over larger domains. Other components of the interTwin DTE have been leveraged to improve the functionality and accessibility of the *normflow* DT (see **Figure 1** and previous WP7 deliverables for more details).

Table 1 – User stories for DT Application: Lattice QCD Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref N | As a | I want to | So that | And it’s considered done when | MoSCoW |
| 4.1-1 | User performing Monte Carlo (MC) simulations | Be able to retrieve and transfer lattice configurations between different HPC systems using a Data Lake. | The user can restart the simulation in a different machine or perform data analysis. | The user can access configurations in a lattice data lake using the toolkit developed by interTwin WP5. | Must |
| 4.1-2 | User performing model training. | Be able to train a generative model for a given field theory, in particular with the method of Normalizing Flows. | Field configurations for various quantum field theories can be generated relatively cheaply. | The model is so well trained that the efficiency is comparable with traditional methods of generating field configurations. To quantify the efficiency, one needs to measure the autocorrelation in the generated configurations. | Must |
| 4.1-3 | User performing data analysis. | Generate field configurations using a trained model. | Various observables of interest relevant to physics applications can be calculated. | The accuracy of the observables matches the desired level. | Must |

## DT Application: Detector simulation

In Task 4.2 a DT application for particle detector simulation has been designed and is being developed.

A methodology that accelerates particle detector simulations by leveraging generative deep learning methods has already been described and is available in D7.6 [[**R9**](#_References)]. Our methodology uses Geant4 [[**R10**](#_References)], a software toolkit for the simulation of the passage of particles through matter, and Generative Neural Networks (GNNs): Generative Adversarial Networks (GAN) and Normalizing Flows (NFs) models. The technical requirements have been identified, defined, and reported in detail in D7.2.

This section outlines the user stories that define the key functionalities and requirements for our Geant4 together with GAN and NF DT Application. These user stories have been identified to reflect the needs of DT operators, including physicists, data scientists, and machine learning engineers. Each user story represents a specific goal from the perspective of the interested stakeholder to guide the development process.

Table 2 – User stories for DT Application: Detector Simulation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref N | As a | I want to | So that | And it’s considered done when | MoSCoW |
| 4.2-1 | DT operator | use the Geant4 application to simulate particles passing through a specific detector setup (full/MC-based simulation). | the operator can generate data for various scenarios | the system successfully simulates particles passing through the specified detector setup, and generates and stores simulation data for further usage. | Must |
| 4.2-2 | Data Scientist | preprocess the simulated data. | it can be used to train a generative model. | the data scientist has access to the raw simulation data, the system allows for data preprocessing and preparation steps, and then the preprocessed data is suitable for model training. | Must |
| 4.2-3 | Software Engineer | train a model on the preprocessed simulated data, with specified model input conditions (e.g. particle’s entrance angle, initial energy and type). | the model can generate data that is similar to the original simulated data. | the machine learning engineer can access and input the preprocessed data into the model, the DT provides tools for monitoring and tuning the training process, and additionally the system validates the trained model by providing performance metrics. | Must |
| 4.2-4 | Physicist | use the trained model during the inference step (fast/ generative network-based simulation). | they can produce generative network-based simulation data faster, in contrast to using traditional Geant4 simulation. | the physicist can use the DT tools or import the trained model into the Geant4 application, the system successfully generates generative network-based simulation data when given initial conditions (e.g. particle’s entrance angle, initial energy and type), and then compares generative network-based simulation data with traditional Geant4 simulation data for consistency and speed. | Must |

## DT Application: Noise simulation for radio astronomy

Task 4.3 is developing a DT of an astronomical source-telescope system, able to generate synthetic output signals identical to the data recorded by a real telescope, which includes both scientifically valuable data and various interference and noise signals. The main approach to build the DT is physics-based, where a set of the control parameters allows adjustment of the output to various sources, detection instruments, and observing conditions. An alternative method of mimicking the available real data based on the geometry of images and noise characteristics is also explored. The resulting data is to be used to train ML data-classification tools. A detailed overview of the project has already been provided in the D7.2 [[**R9**](#_References)] and recently updated in D7.8 [[**R1**](#_References)].

The work is split into three parallel and interacting subprojects: astrophysical analysis of the real data, theoretical modelling of the source/telescope system, and development of a fast and scalable C++ implementation. The first of these includes building of the ML-data classification tool for the analysis of the real data, which assigns labels to each data fragment based on the type of signal detected or not detected in it. The label describes the fragment on a basic level as “scientifically important data”, “no signal”, “interference” and similar, and, in the future, may also include more detailed properties. Since the proportions of each data type in the real data flow are very different (e.g. scientifically important data might constitute less than 1% of the data sample), efficient ML training requires synthetic data to be used. That is where the DT comes in, which is developed in the second subproject based on a physical model of the source, its signal transmission and registration. Finally, within the third subproject, all the tools are combined in an easily deployable container, suitable to run efficiently on modern HPC systems.

Table 3 – User stories for DT Application: Noise Simulation for Radio Astronomy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref N | As a | I want to | So that | And it’s considered done when | MoSCoW |
| 4.3-1 | Radio Telescope Operator / On-Site Radio Astronomer. | get DT- generated synthetic data tailored to the specific observation type, target and conditions, and train the ML data classifier with them. | the ML data classifier can be used in flagging the scientifically worthless data during the observation run to keep the recorded data volume as low as possible. | DT-trained ML data classifier labels the real data by type (science, empty, interference etc.)  with a high degree of certainty (~95%). | Must |
| 4.3-2 | Radio Astronomer (responsible for processing and assessing the data). | be able to run DT- generated synthetic data through data processing pipelines and analytic tools. | the pipelines and tools can be debugged and correctly configured prior to the arrival of real data, improving the efficiency and shortening the time before the data release. | there is no apparent difference when running the synthetic data through the relevant pipelines and tools. | Must |
| 4.3-3 | Radio Astronomer (scientific analyst, “end user”). | use the DT- generated and processed data. | hypotheses about the real data can be tested. | the synthetic data is tried in a scientific analysis of a real project, and the end users are happy with the results. | Should |
| 4.3-4 | Radio Astronomer or Software Engineer (data acquisition/ processing pipeline developer). | run the DT and ML data classifier training in parallel configuration on computing clusters. | run time can be decreased to achieve (near) real-time data processing. | near real-time data processing is achieved. | Should |

## DT Application: VIRGO Noise Detector

The goal of Task 4.4 is to produce a DT of the Advanced Virgo interferometer to realistically simulate transient noise in the detector. We are currently using GNNs to determine the relationship between *strain* data (that measures the deformation induced by the passage of a Gravitational Wave (GW)) and *auxiliary* data (that monitors the status of the detector’s subsystems as well as the environmental conditions). The trained model will be used in a pipeline for vetoing and denoising the strain signal in low-latency searches, i.e. those data analysis pipelines that search for transient astrophysical signals on shorter timescales and in almost real time. The high-level architecture of the DT and its implementation as a series of modules orchestrated by Airflow has been defined in D7.8 [[**R1**](#_References)]. The DT users for this application are the expert users operating the vetoing/denoising pipeline, as well as people working in the Rapid Response Team on shift during the observing period. The other stakeholders are the *physicists* operating the *downstream pipelines* that will use the information provided by the DT. In **Table 4**, for each of the two stakeholders, a list of requirements that drive the design of the DT is reported.

Table 4 – User stories for DT Application: VIRGO Noise Detector

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref N | As a | I want to | So that | And it’s considered done when | MoSCoW |
| 4.4-1 | DT Operator | make sure that the GNN model is periodically re-trained on most recent data. | the DT realistically simulates the detector response following any change in the experimental conditions. | the model re-training converges and has a good accuracy in reproducing the flux of incoming data. | Must |
| 4.4-2 | DT Operator | make sure that the DT is able to identify transient noise (glitches) in incoming data. | the information about an identified glitch can be used to issue a veto decision. | the DT outputs a probability for a given time span of data to contain a glitch. | Must |
| 4.4-3 | DT Operator | make sure that the DT is able to reproduce transient noise (glitches) in incoming data. | the information about an identified glitch can be used to denoise the incoming signal. | the DT outputs a signal in which the glitch has been removed (denoised). | Should |
| 4.4-4 | DT Operator | make sure that the DT delivers the correct veto flag to downstream pipelines. | downstream pipelines can use this information to decide if further processing of the data or not. | the DT delivers to downstream pipelines a veto decision in the expected format. | Must |
| 4.4-5 | DT Operator | make sure that the DT is able to denoise the incoming signal. | downstream pipelines can search for a GW signal in incoming data without being biased by glitches. | the DT delivers to downstream pipelines a stream of denoised data in the expected format. | Should |
| 4.4-6 | Physicist | be able to use the DT veto information in downstream pipelines. | data containing glitches are discarded from processing. | the information about the probability for data to contain a glitch is received in the expected format. | Must |
| 4.4-7 | Physicist | ensure that downstream detection pipelines receive an input of denoised data. | the search for a GW signal is unbiased by glitches. | the stream of denoised data is received in the expected format and the denoising procedure has not removed any astrophysical signal. | Should |

# DT Applications Design

## DT Application: Lattice QCD simulation

Lattice QCD aims to shed light on the properties of QCD in the low energy limit where perturbation theory breaks down and numerical approaches are required. In interTwin the objectives are twofold: a classical scenario looking at how new tools, developed within interTwin, can facilitate the data flexibility needed by modern international collaborations, and a second more speculative ML-based scenario, looking at demonstrating ML-assisted simulations at the proof-of-concept level.

### Advanced data management for Lattice QCD

Lattice simulations are executed at large scale on HPC systems controlled by a batch system. In order to facilitate data analysis, lattice data should be readily available to the members of a collaboration in a controlled way. Within interTwin, it was agreed that one path to better data management in the scientific context involves the use of federated identities and group-based access control. Thanks to the efforts of WP5 and the ILDG, there is now a prototype functional “Lattice Datalake” with three storage endpoints (SEs), two at DESY and one at CESGA. **Figure 2** illustrates the flow of control when accessing the datalake, and highlights how a user, once authenticated, could also access the ILDG’s FC and MDC.

In order to access the datalake and its SEs, one needs to register with the ILDG and then configure an oidc-agent [[**R11**](#_References)] on their computer to interact with the ILDG’s public client [[**R12**](#_References)]. The oidc-agent and public client handle the token exchange required by the ILDG’s Identity and Access Management (IAM) service for authentication and authorisation. The Access Control Service (ACS) in Figure 2 represents a policy decision point that, among other functions, acts as a security mechanism, protecting resources from unauthorized access. Exactly what permissions a user has is controlled by the IAM/ACS and encoded in the exchanged tokens. ILDG’s IAM is flexible with respect to permissions. This is useful as a typical lattice collaboration will divide tasks between members in such a way that it is desirable to, for instance, restrict write/modify access from all users other than those directly responsible for lattice generation runs.



Figure 2 - Schema depicting the flow of control when accessing the Lattice Datalake. In this example the institutional IdP is CSIC, but it could be any other IdP trusted by the ILDG.

Once a user has gained access to the datalake, they can query the SEs and start transferring (subject to their permissions) data between them. rclone [[**R13**](#_References)] is a useful tool for querying and transferring small files. For larger file transfers it is necessary to do third party copies, which rclone cannot do. Although it is possible to do third party copies with the curl command, it is far preferable to use the File Transfer Service (FTS) provided by EGI through its FTS server at CERN [[**R14**](#_References)]. The FTS can handle large numbers of file transfers efficiently and supports multiple transfer protocols. It lets users queue multiple requests while balancing loads to avoid bottlenecks and the overwhelming of network/storage systems. In this way, it serves a role analogous to the batch systems used on HPCs to ration computer time.

The lattice datalake is currently being benchmarked by timing the transfer of data between its SEs. Demoing the transfer of real lattice data to the CESGA SE is one of Task 4.1’s near-term goals. Having a functional lattice datalake is also an opportunity to experience and explore first-hand the possibilities afforded by ILDG’s IAM/ACS. Plans for the future include placing an additional Rucio layer in between the FTS and the user for added security and stability. A datalake user would then only need to use rucio commands to interact with the datalake. For a discussion of the potential paths towards ILDG-Rucio integration, see the previous deliverables [[**R15**](#_References)]. This will require further ILDG-Rucio-interTwin collaboration. But even in the absence of Rucio integration the nascent Lattice Datalake is a compelling demonstration of the utility of a federated lattice identity.

### Generative models using Machine Learning

Machine learning techniques are being explored in Lattice QCD in order to facilitate the generation of configurations in complex areas of the parameter space. In this respect, the training of the models is done by comparing the result of the ML technique with the result of a standard MC simulation.

The efficiency of general purpose MC algorithms decreases dramatically when the simulations need to take place near critical points due to critical slowing down. This is a general phenomenon in simulations in Physics related to phase transitions, which happens as well in Lattice QCD, for example with simulations at very fine distances that are needed for extrapolation to the continuum limit. Simulations need to take place in areas of the parameter space where topology freezing (among other factors) induces very large autocorrelations.

Whether or not ML can speed-up the field configuration generation in those parts of the parameter space is a subject under investigation. A series of recent studies suggest that using Normalizing Flows (a class of deep generative models) may help to improve this situation (a block diagram illustrating the method is shown in **Figure 3**). The underlying idea is to use ML techniques to map the theory of interest to a “simpler”, easier to simulate theory. This approach has the potential to become more efficient than traditional sampling especially when the concept of transfer learning is utilised. However, the costs associated with the (highly complex) sampling from the path integral, are transferred to the training of a model. The question under investigation is therefore how expensive it is to train an ML model compared with making a classical MC simulation.

A diagram of a system

Description automatically generated

A diagram of a flowchart

Description automatically generated

Figure 3 - Upper part: Graphical representation of the classical generation of configurations using MC algorithms; Lower part: Graphical representation of the Normalizing Flows method including a correcting accept/reject step to account for the fact that the model cannot be perfectly trained

The purpose of this work is designing better architectures for ML models so that the acceptance rates become reasonable (~50% or more) as the volume of the lattice increases. The requirements in terms of resources are not as in the classical MC simulation since the methodology is still at the proof of concept level.

T4.1 has been developing the normflow [[**R6**](#_References)] package to study and experiment with ML-assisted lattice generation. Normflow supports scalar theories and was recently extended to accommodate gauge theories, broadening its applicability. In a nutshell, three essential components are required for the method of normalizing flows:

* A prior distribution to draw initial samples.
* A Neural Network (NN) to perform a series of invertible transformations on the samples.
* An action that specifies the target distribution, defining the goal of the generative model.

The central high-level class of the package is called *Model*, which can be instantiated by providing instances of the three objects mentioned above: the prior, the NN, and the action. To specify the theory, the user can choose from the available actions in the package, such as a quartic scalar action or the Wilson gauge action. Next, an appropriate network can be assembled using the package's modules, which automatically calculate the Jacobian of the transformations. Similarly, an appropriate prior distribution can be selected, for example, a Gaussian distribution for scalar theories or uniformly generated SU(N) matrices (with the Haar measure) for gauge theories. For a quick start, please refer to the examples in the normflow repository at [**https://github.com/interTwin-eu/Use\_Case\_T4.1\_normflow/tree/normflow\_public\_v1.1/examples**](https://github.com/interTwin-eu/Use_Case_T4.1_normflow/tree/normflow_public_v1.1/examples).

Each instance of *Model* comes with a *train* method, responsible for training the model. The training is based on a self-learning strategy, meaning no external data is required to train the model. The goal is to optimize the NN's parameters to accurately map the prior distribution to the target distribution. The training process begins by generating samples from the prior and feeding them into the NN. The output is then used to compute the Kullback-Leibler (KL) divergence, which serves as the default loss function, and the model aims to minimize this loss. KL divergence is a common measure of how one probability distribution diverges from a second distribution. In this case, it quantifies the difference between the distribution of the transformed data (i.e., the model's predictions) and the target distribution. In addition to KL divergence, the train method also supports alternative optimization strategies, such as maximizing the effective sample size (ESS).

In KL divergence minimization, the total derivative of the loss decomposes into a partial derivative with respect to the parameters and the transformed variable. While the latter contribution remains, the partial derivative with respect to the parameters statistically vanishes. To enhance stability, the package applies a reverse flow correction [[**R16**](#_References)], which removes these statistically vanishing terms by default. However, it can be disabled, which speeds up training by roughly a factor of two per epoch, at the cost of reduced training effectiveness.

The calculation of the KL divergence requires computing the determinant of the Jacobian matrix of the transformation. To facilitate this, the package defines an abstract class called *Module\_*, which is a subclass of *torch.nn.Module*. The trailing underscore in the class name indicates that the *forward* method not only returns the transformed inputs but also computes and returns the logarithm of the Jacobian determinant as the second item in a two-item tuple. By encapsulating the calculation of the determinant of the Jacobian matrix, the *Module\_* class provides a structured way to handle transformations and their inverses efficiently, making it suitable for use in optimization processes. Examples in the normflow repository demonstrate how models can be constructed using various subclasses of *Module\_* and then trained with the methods inherited from *torch.nn.Module*.

## DT Application: Detector simulation

In D7.6 [[**R9**](#_References)], the underlying challenges of detector simulation for CERN and the HEP community, as well as the importance of developing a DT digital twin system that integrates simulation methods with ML, were analysed and described.

This section provides a comprehensive overview of CERN’s DT application of a detector simulation. It describes the key steps, from particle simulations to event generation, and subsequent data comparison with real data. The process is explained in detail, highlighting the functionalities at each stage. Furthermore, it illustrates the flexibility in tuning the system to accurately represent various detectors’ responses. This explanation is designed to give readers an understanding of the entire workflow design, shedding light on current practices and potential areas of future improvement. It also opens the way for a deeper discussion on the challenges faced, decisions made, and future strategies in the ongoing development of this innovative simulation application.

This application consists of two components: the component that incorporates the Geant4-based simulation framework and the deep learning component, which uses deep generative models based on a specified particle detector setup. The two components are encapsulated into two main workflows, the training workflow and the inference workflow, as illustrated in **Figure 4**. Below, the application functionalities and their specifications included in each workflow are described.

A computer screen shot of a diagram

Description automatically generated

Figure 4 – Fast particle detector simulation using ML techniques high level workflow composition and its connections with other work packages’ components

The Geant4 simulation toolkit that consists of an important component of CERN’s application, performs particle physics simulations based on MC methods. It constitutes a set of components which include geometry and tracking descriptions, detector response modelling, event management, user interfaces and many other functionalities. Geant4 toolkit is typically used in HEP research projects for complex detectors of which single components (i.e. the calorimeters) are simulated using generative models, as an alternative to the classical MC techniques. Calorimeters are key components of the whole experimental setup, which are responsible for measuring the energy of the particles. Simulating the calorimeters’ response using Geant4 is usually a bottleneck for the related research projects. For that reason, generative AI based fast simulation is being leveraged, which generates directly the detector output, without reproducing, step by step, each single particle that interacts with the detector material, in contrast to MC methods.

The training workflow design includes the following functionalities, which run on HPC systems managed by Kubeflow containerized components. Geant4 simulates particle interactions, producing data based on a detector-specific configuration. The produced data consists of the energy measured by the detector sensors, the properties of the initial particle, such as its type, energy, and its trajectory angle with respect to the detector volume, and other metadata. The produced data, in ROOT [[**R17**](#_References)] format, is stored at different data centres, with CERN currently serving as the primary storage site. The data requires conversion into the Hierarchical Data Format version 5 (HDF5) for further preprocessing before being input into the generative model. This conversion is performed using a Python script. Following the ROOT to HDF5 format conversion, the HDF5 data is further preprocessed and transformed into PyTorch arrays, a process currently incorporated within the model training scripts.

A GAN or a NF model CaloINN model [[**R18**](#_References)] is trained on the preprocessed data, conditioned on specific input describing the properties of the particles. The data is retrieved from the storage space where they reside. Hyperparameter optimization (HPO) is also employed to improve model performance. During validation and HPO, the model generated data and the Geant4 simulated data distributions are both visualised. Training, validation and HPO processes run on HPC systems. The following functionalities are currently implemented for the GAN model, and not for the NF model.

The training workflow stores the optimised models, selected based on validation results, and converts them into the ONNX format for use during inference. Transformation of the model architecture and weights is performed within a Python script. The model registry is managed by Task 6.5.

The inference workflow is needed to generate calorimeter signals using the trained model. The workflow runs on HPC systems managed by Kubeflow containerized components. The Geant4 application at this stage initiates a particle, guiding it through the detector until it reaches the bottleneck detector part (the calorimeter), at which the ML model performs inference. The model's output undergoes a detector-specific transformation to convert it into a Geant4 suitable input, the 3D images that the model generates are mapped into the so-called "hits" data consisting of the position (x, y, z coordinates) in the detector (i.e. the sensors positions) and the corresponding energy measurements.

The transformed data can be used by the Geant4 framework to complete the process of generating events, simulating the passage of particles through the remaining components of the detector. Data distribution comparisons are drawn between the ML-generated data and real data (either derived from a traditional Geant4 simulation or data derived from accelerator test beams). These comparisons are essential for validating the efficacy and accuracy of the ML-generated data.

Finally, based on the results visualised, two possible workflows are proposed for simulation tuning, shown in **Figure 5**. Within the inference workflow, a model can be re-inferred with different model input parameter values, provided these parameter values have been accounted for during model training. Alternatively, if a different value range of the conditional parameters is needed, the training workflow must be re-run from the beginning. These two possible workflows allow for greater flexibility and adaptability in tuning the detector's responses to various particle interactions.

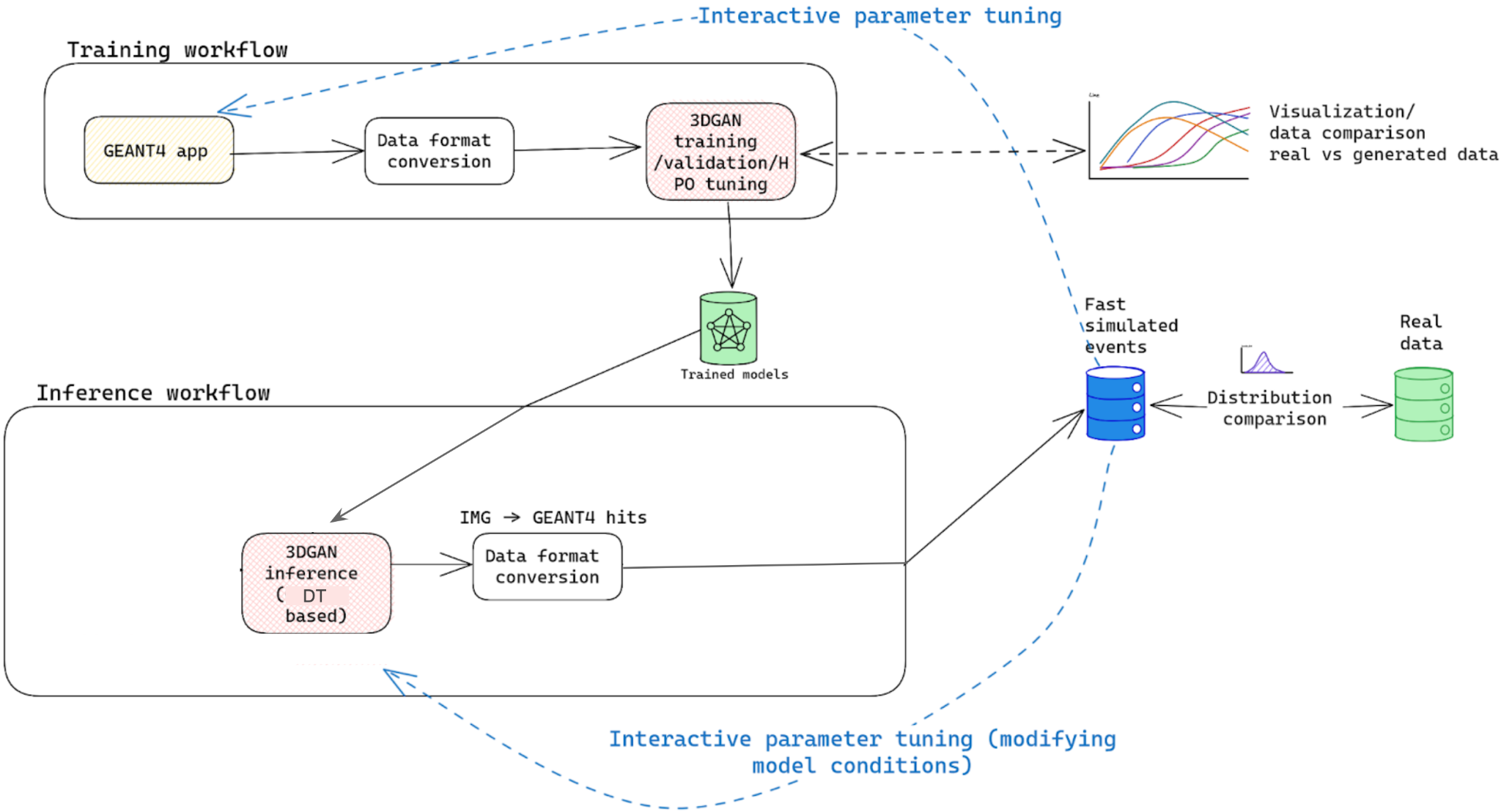
**

Figure 5 - Detailed graph representation of the training and inference workflows composition (as described above) of the fast particle detector simulation DT utilising 3DGAN approach

## DT Application: Noise simulation for radio astronomy

This DT application addresses the challenge of identifying radio signals from intermittent astrophysical sources, so called ‘transients’ and ‘pulsars’, from large-volume data streams during the data acquisition phase. One of the main tasks is to identify noise and interference signals coming from different sources. The DT recreates the propagation of pulsar signals from the source to radio astronomical antennas and processing by radio telescope electronics (see **Figure 6**), generating synthetic output signals identical to the data recorded by real telescopes.

The DT is a part of the development effort of a larger framework called ML-PPA for Machine Learning-based Pipeline for Pulsar Analysis [[**R19**](#_References)] (see **Figure 7**). In addition to the DT, it includes a Convolutional Neural Network (CNN)-based ML classifier of the pulsar data, which can be trained using the DT-supplied data. An essential part of this project is analysis of real astronomical data (collected by the Effelsberg [[**R20**](#_References)] and MeerKAT [[**R21**](#_References)] radio telescopes observing a number of bright and well-studied pulsars) and creation of empirical simulated data based on it (i.e. empirical-based DT), which provides the material for testing both the ML-classifier and the physics-based DT.

A diagram of a radio station

Description automatically generated

Figure 6 - General outline of the DT structure: modelling the astrophysical source (pulsar) within the “light-house” model [[**R22**](#_References)], transmission of the signal through the interstellar matter, receiving and processing by a radio telescope, adding sources of both natural and artificial interference and noise, finally resulting in frequency-time graphs similar to those obtained with real radio telescopes

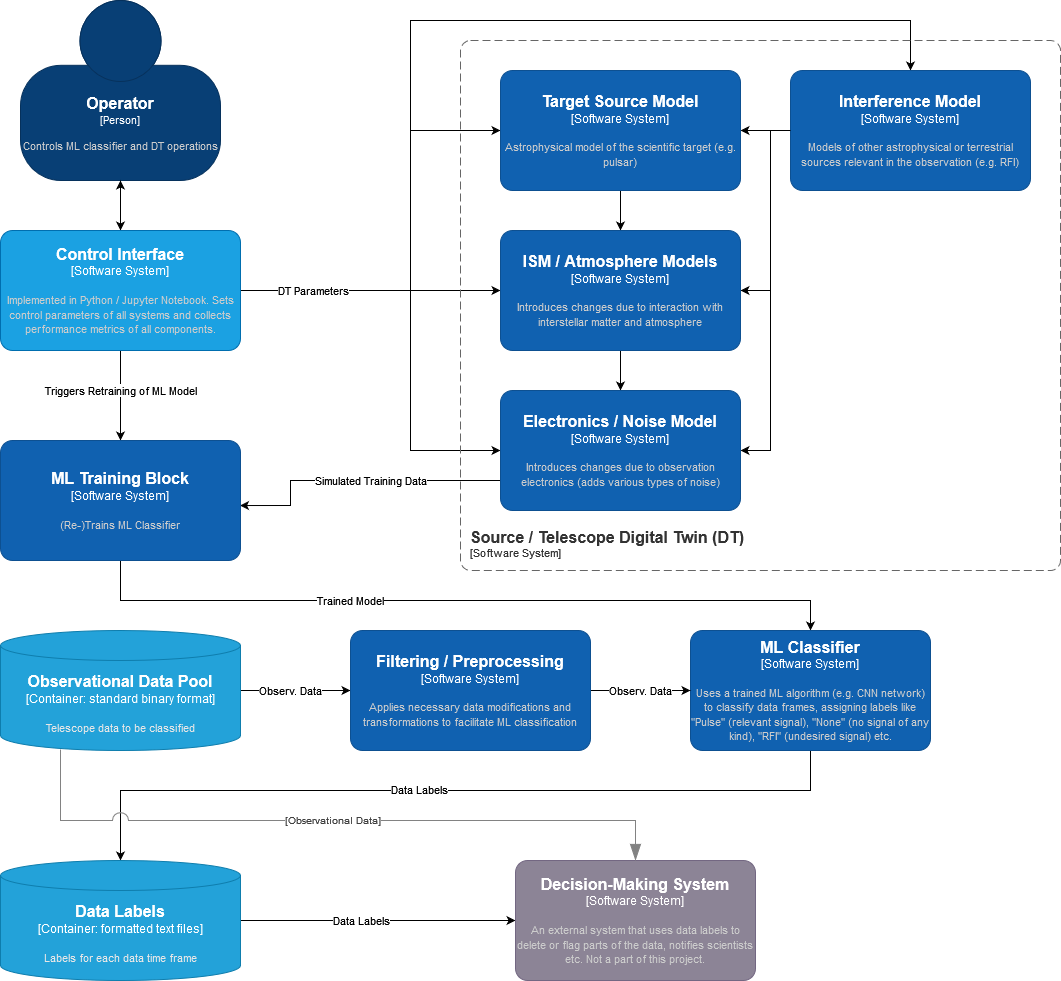


Figure 7 - Diagram of the ML-PPA (in the C4 model)

The overall software architecture follows a 3-layer design (**Figure 8**). The top layer provides interfaces for the user to develop pipelines, which are developed in WP4, by combining tools and algorithms that are held in the middle layer, which has already been documented in D7.2 [[**R6**](#_References)] and D7.8 [[**R1**](#_References)]. The bottom layer enables the creation of containers that can be distributed to data centres via the integration of Workflow tools developed in WP6, where the built-in pipelines can be used to analyse or generate data. For logistical reasons, most components are first developed and tested in Python, and then some parts are rewritten in C++ to ensure the best speed and efficiency.

The development also addresses the issue of poor scalability of available radio astronomical software tools, the aim is to create a package that can be efficiently used for massively parallel computing over the resources offered by the WP5 infrastructure.

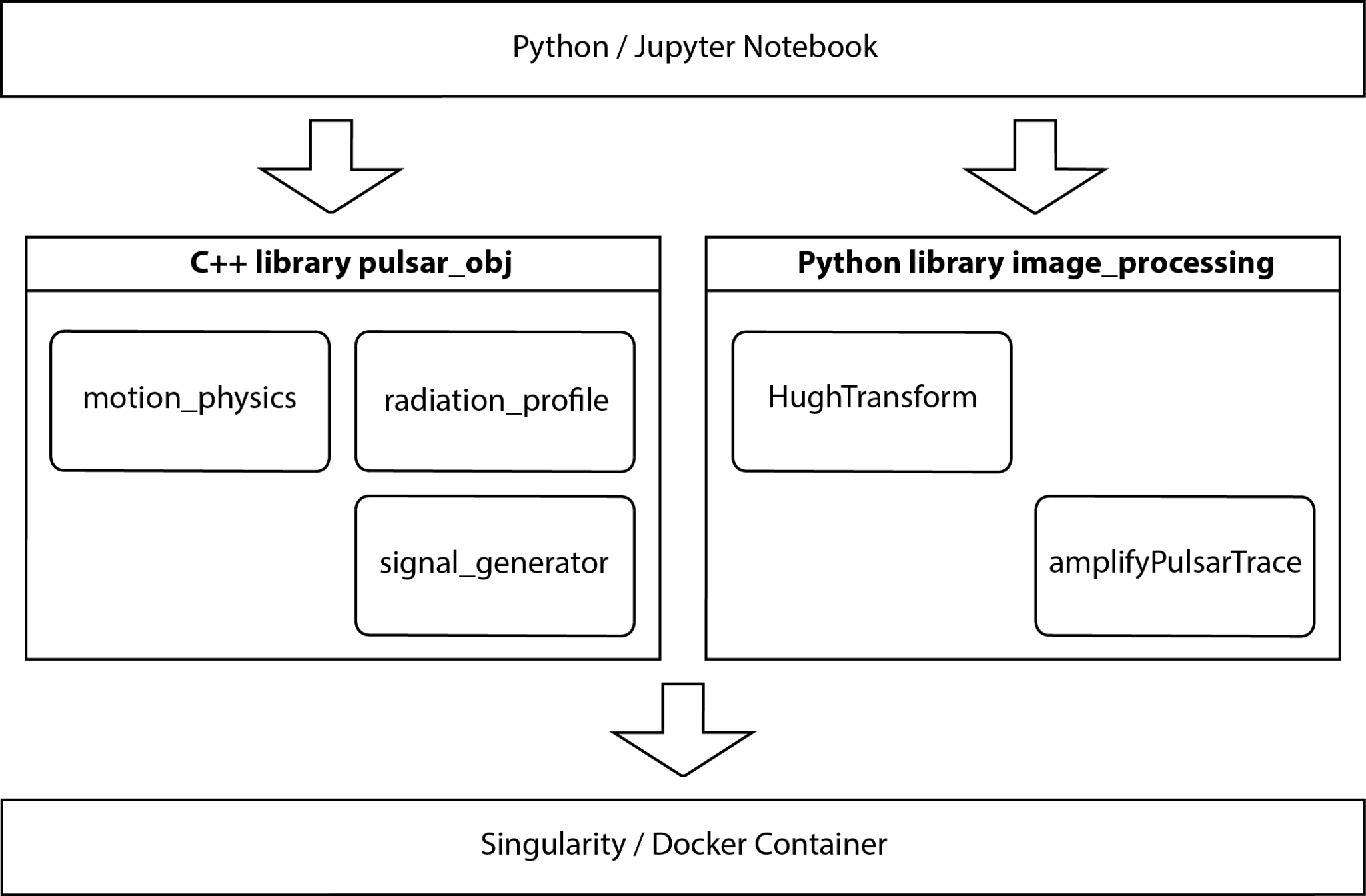


Figure 8 - Layered software architecture of the framework ML–PPA

The ultimate future goal of this framework beyond this project is to empower astronomers in their pursuit of uncovering non-trivial astronomical signals and enhancing their ability to process, analyse, and interpret huge volumes of data coming from the next generation of radio telescopes, such as Square Kilometer Array (SKA) [[**R23**](#_References)] "pathfinders", like the above-mentioned MeerKAT or Australian ASKAP [[**R24**](#_References)], and then the SKA itself, when it is available online.

## DT Application: VIRGO Noise detector

The purpose of Task 4.4 is to develop a DT application that can simulate glitches in the observational channel of the Virgo GW interferometer [[**R25**](#_References)] from its control channels.

A glitch is a transient noise artefact observed in the observational (*strain*) channel. Glitches can occur due to environmental reasons, such as seismic activity, or as a result of resonances in the detector’s subsystems. The monitoring of such effects is entrusted to a high number of control sensors, whose continuous data output is mapped onto so-called *auxiliary channels*. Glitches are divided into the classes identified by the GravitySpy project [[**R26**](#_References)].

The high number of observed glitches in the past observational runs and in the current O4b [[**R27**](#_References)] run makes it necessary to put into place effective vetoing procedures (**Figure 9**) for the identification and removal of bad data from the analyses’ pipelines. For this reason, the project aims to identify which auxiliary channels correlate strongly with the glitches observed in the strain and use generative methods to map the glitch from the former to the latter.

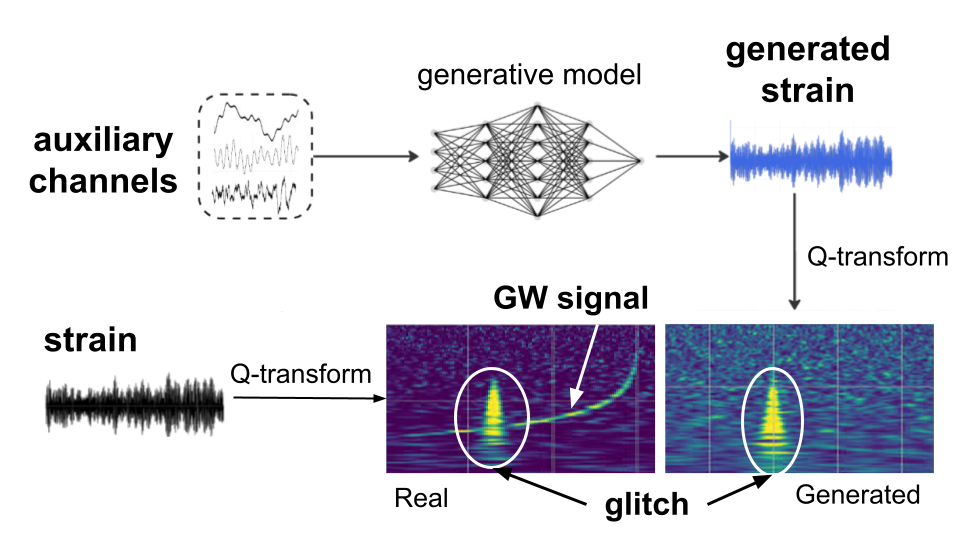


Figure 9 - The Virgo DT schema. Data from the auxiliary channels are used to train the GNN . The spectrograms of both generated and measured strain data are compared to show the transient noise glitch

This release of the DT consists of an update on the train and inference subsystems presented in the previous release [[**R28**](#_References)]. The modules that are present are implemented as a collection of Jupyter notebooks and installable Python packages via pip.

The two main packages presented in this deliverable are **ANNALISA**, an installable Python package, and **GlitchFlow**, presently in the form of a Jupyter notebook. They are the main components of the training subsystem of the DT. The former is used for identifying the relevant auxiliary channels which the NN will use as input, while the latter is used for the generation of glitches in the strain channel starting from the input data. The inference subsystem is instead organised in two different Jupyter notebooks, Preprocess\_API, used for data preprocessing and dataset creation and Generative\_API, for glitch generation. Preprocess\_API is also used in the training subsystem.

**ANNALISA** (Advanced Nonlinear transient-Noise Analyser of Laser Interferometer Sensor Arrays) is a tool that makes use of time-frequency domain analysis of the data, namely the q-transform, to evaluate correlations among the main and auxiliary channels as the ratio of temporally coincident spikes in the energetic content of the signals above a critical threshold over the total number of spikes in the main channel. The current version of ANNALISA employs a PyTorch-implemented q-transform which we developed to run the whole analysis on GPU. Our version of the q-transform is equivalent to the GWPy [[**R29**](#_References)] implementation up to some border effects; this development makes it possible to speed up the correlation analysis by two orders of magnitude.

**GlitchFlow** is the module that contains the GNN for generating the glitches. In this new implementation, the NN architecture consists of a U-net with residual blocks; the skipped connections of the U-net are enhanced by attention gates. It was found that this new implementation performs significantly better than the previous ResNet12, allowing us to achieve a vetoing accuracy of 91% over scattered light glitches, which constitute the most common class of glitches.

The training process involves feeding the generative model in GlitchFlow with data from the auxiliary channels identified by ANNALISA and the corresponding glitches observed in the main channel. Through iterative training, the model learns to capture the complex relationships and dynamics present in the data, allowing it to generate realistic glitches that accurately reflect the characteristics of the observed glitches.

Since the project aims to implement a vetoing protocol to flag portions of the datastream as corrupted, it is of paramount importance to avoid the erroneous characterisation of GWs as glitches. The only way to achieve this is to provide as input to the NN only channels which do not detect any astrophysical signals; these channels are termed *safe channels*. A list of safe channels is available within the Virgo collaboration. This classification is constantly being updated for each new observational run, as channels are replaced, or their technical specifications are changed.

With tens of thousands of safe channels in Virgo, feeding all of them into the analysis tool would be unfeasible. Instead, we leverage the understanding of specific glitch phenomena, such as scattered light glitches, which are currently the most frequent type observed. These glitches occur due to microseismic activity coupling with the detector, leading to photon scattering on moving reflective surfaces. By considering the physical origins of these glitches and their detection by sensors monitoring optical bench movement, velocity, or acceleration, relevant input channels possessing the expected units of these quantities are identified. Channels monitoring the global status of the interferometer’s optical components are also considered. This approach ensures a more precise and effective channel selection process and makes it possible to restrict the correlation scan to a few hundred auxiliary channels.

The channels selected by the ANNALISA module, i.e. those showing a correlation coefficient above a tunable threshold, are then passed to Preprocess\_API in order to create the dataset to train the GNNs in GlitchFlow for the task of glitch generation.

*A diagram of a data processing process

Description automatically generated*The high-level architecture of the DT has been defined in D7.6 section 2.3.2 [[**R9**](#_References)]. We show here the C4 model of the DT veto pipeline (**Figure 10**). This was also already presented in deliverable 7.2 and it is repeated here for clarity.

Figure 10 - System Context diagram (in the C4 model) of the DT for the veto pipeline

# Conclusions

The final version of the design of interTwin DTs applications for WP4 concerned with the physics domain was developed during the first three years of the project. In this deliverable, the main focus was on defining the final versions of the designs of the different modules and presenting the utility of these modules to the main beneficiaries. As part of each task (T4.1, T4.2, T4.3, and T4.4), user stories, the individual outcomes, and the steps that were taken to accomplish their specific goals are described. Additionally, in Section 3, we outlined the layout of each individual digital twin.

The next step will be to finalise the integration of each DT Application with its DTE, this will be described in the final deliverable 4.8 (D4.8), planned for July 2025.

# References

|  |  |
| --- | --- |
| Reference | |
| **No** | **Description / Link** |
| **R1** | Miranda, E. "Time boxing planning: buffered moscow rules." *ACM SIGSOFT Software Engineering Notes* *36.6* (2011).  DOI: [**https://doi.org/10.1145/2047414.2047428**](https://doi.org/10.1145/2047414.2047428) |
| **R2** | Gaurav Sinha, R. et al.“interTwin D7.8 Final version of the thematic modules for the physics domain” . *Zenodo.* (Version 1 Under EC Review). (2025).  DOI: [**https://doi.org/10.5281/zenodo.14931996**](https://doi.org/10.5281/zenodo.14931996) |
| **R3** | Spiga, D. et al. “interTwin D5.4 Final DTE Infrastructure Software Release”. (V1 Under EC review). *Zenodo.* (2025).  DOI: [**https://zenodo.org/records/14727089**](https://zenodo.org/records/14727089) |
| **R4** | Kobyzev, I.; Prince S. and Brubaker, M.A. **“**Normalizing Flows: An Introduction and Review of Current Methods”.*IEEE Transactions on Pattern Analysis and Machine Intelligence* *( Volume: 43, Issue: 11)*. ( 2021).  DOI**:** [**https://doi.org/10.1109/TPAMI.2020.2992934**](https://doi.org/10.1109/TPAMI.2020.2992934) |
| **R5** | Beckett, M. G., et al. "Building the international lattice data grid." *Computer Physics Communications* *182.6* (2011).  DOI: [**https://doi.org/10.1016/j.cpc.2011.01.027**](https://doi.org/10.1016/j.cpc.2011.01.027) |
| **R6** | Komijani, J. and Ray, G. normflow\_ (Version v1.1.0-beta) [Computer software]. (2024).  [**https://github.com/jkomijani/normflow\_**](https://github.com/jkomijani/normflow_) |
| **R7** | Komijani, J. and Marinkovic, M.K. “Generative models for scalar field theories: how to deal with poor scaling?” . *The 39th International Symposium on Lattice Field Theory*. (2022).  DOI:[**https://arxiv.org/abs/2301.01504**](https://arxiv.org/abs/2301.01504) |
| **R8** | Javad Komijani, J. and Marinkovic, M.K. “Normalizing flows for SU(N) gauge theories employing singular value decomposition”*.* *The 41th International Symposium on Lattice Field Theory*. (2024).  DOI:[**https://arxiv.org/abs/2501.18288**](https://arxiv.org/abs/2501.18288) |
| **R9** | Vallero, S. et al. “interTwin D7.6 Updated report on requirements and thematic modules functionalities for the physics domain“. *Zenodo.* (2024).  DOI: [**https://zenodo.org/records/14975072**](https://zenodo.org/records/14975072) |
| **R10** | Agostinelli, S. et al. "GEANT4—a simulation toolkit." *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment* *506.3* (2003).  DOI: [**https://doi.org/10.1016/S0168-9002(03)01368-8**](https://doi.org/10.1016/S0168-9002(03)01368-8) |
| **R11** | “How OpenID Connection Works”. Accessed on 17/03/2025: [**https://openid.net/developers/how-connect-works/**](https://openid.net/developers/how-connect-works/) |
| **R12** | “FST client- installation and configuration manual”. Accessed on 07/03/2025: [**https://confluence.egi.eu/display/interTwinDTE/FTS+client+-+installation+and+configuration+manual**](https://confluence.egi.eu/display/interTwinDTE/FTS+client+-+installation+and+configuration+manual) |
| **R13** | Bailleul, D.; Stoeckel, S. and Arnaud‐Haond, S. "RClone: a package to identify MultiLocus Clonal Lineages and handle clonal data sets in r." *Methods in ecology and evolution* *7.8* (2016).  DOI: [**https://doi.org/10.1111/2041-210X.12550**](https://doi.org/10.1111/2041-210X.12550) |
| **R14** | Murray, S. et al. "FTS Service Evolution and LHC Run-3 Operations." *EPJ Web of Conferences*. *Vol. 295. EDP Sciences* (2024).  DOI: [**https://doi.org/10.1051/epjconf/202429501031**](https://doi.org/10.1051/epjconf/202429501031) |
| **R15** | Vallero, S. et al. “interTwin D7.6 Updated report on requirements and thematic modules functionalities for the physics domain”.  *Zenodo.* (1 Approved by the EC).(2024).  DOI: [**https://zenodo.org/records/14975072**](https://zenodo.org/records/14975072) |
| **R16** | Vait, L.l et al. "Gradients should stay on Path: Better Estimators of the Reverse- and Forward KL Divergence for Normalizing Flows". (2022).  DOI: [**https://arxiv.org/abs/2207.08219**](https://arxiv.org/abs/2207.08219) |
| **R17** | “ROOT: analyzing petabytes of data, scientifically.” Accessed on 21/03/2025.  [**https://root.cern/**](https://root.cern/) |
| **R18** | Ernst, F. et al. “Normalizing Flows for High-Dimensional Detector Simulations”. (2025).DOI: [**https://arxiv.org/abs/2312.09290**](https://arxiv.org/abs/2312.09290) |
| **R19** | Kazantsev, A. et al. “ML-based Pipeline for Pulsar Analysis (ML–PPA)”  [**https://gitlab.com/ml-ppa/gitlab-profile/-/blob/main/PUNCH\_interTwin\_project.pdf**](https://gitlab.com/ml-ppa/gitlab-profile/-/blob/main/PUNCH_interTwin_project.pdf) |
| **R20** | “Radio Telescope Effelsberg”. Accessed on 17/03/2025: [**https://www.mpifr-bonn.mpg.de/en/effelsberg**](https://www.mpifr-bonn.mpg.de/en/effelsberg) |
| **R21** | “MeerKAT Radio Telescope”. Accessed on 17/03/2025: [**https://www.sarao.ac.za/gallery/meerkat/**](https://www.sarao.ac.za/gallery/meerkat/) |
| **R22** | Lyne, A. G. and Graham-Smith, F. “Pulsar Astronomy”, *Cambridge Astrophysics Series, Cambridge University Press.* (1990).  DOI: [**https://doi.org/10.1063/1.2810218**](https://www.researchgate.net/publication/258734664_Pulsar_Astronomy) |
| **R23** | “SKA Observatory”. Accessed on 17/03/2025.  [**https://www.skao.int/en**](https://www.skao.int/en) |
| **R24** | ASKAP Radio Telescope. Accessed on 17/03/2025: [**https://www.csiro.au/en/about/facilities-collections/atnf/askap-radio-telescope**](https://www.csiro.au/en/about/facilities-collections/atnf/askap-radio-telescope) |
| **R25** | “Virgo”. Accessed on 17/03/2025: [**https://www.virgo-gw.eu/**](https://www.virgo-gw.eu/) |
| **R26** | Zevin, M., et al. "Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science." *Classical and quantum gravity* *34.6* (2017).  DOI: [**https://iopscience.iop.org/article/10.1088/1361-6382/aa5cea**](https://iopscience.iop.org/article/10.1088/1361-6382/aa5cea) |
| **R27** | “LIGO and Virgo detectors restart gravitational wave observation“. Accessed on 17/03/2025: [**https://www.virgo-gw.eu/news/ligo-and-virgo-detectors-restart-gravitational-wave-observation/**](https://www.virgo-gw.eu/news/ligo-and-virgo-detectors-restart-gravitational-wave-observation/) |
| **R28** | [Pidopryhora,](https://zenodo.org/search?q=metadata.creators.person_or_org.name:%22Pidopryhora,+Yurii%22) Y. et al. “interTwin D4.4 First version of the DTs capabilities for High Energy Physics, Radio astronomy and Gravitational-wave Astrophysics“. *Zenodo.* (2024).  DOI:[**https://doi.org/10.5281/zenodo.14974752**](https://doi.org/10.5281/zenodo.14974752) |
| **R29** | Macleod, D. M. et al. "GWpy: A Python package for gravitational-wave astrophysics." *SoftwareX* *13* (2021).  DOI: [**https://doi.org/10.1016/j.softx.2021.100657**](https://doi.org/10.1016/j.softx.2021.100657) |
|  |  |